DOCUMENT RESUME

ED 264 406 CE 043 087

AUTHOR Kittell, Linda; Walczak, Joseph
TITLE Computer Education Curriculum. Connecticut Vocational Technical School System. Version 4.
INSTITUTION Connecticut State Dept. of Education, Hartford. Div. of Vocational-Technical Schools.
PUB DATE Mar 85
NOTE 133p.
PUB TYPE Guides - Classroom Use - Guides (For Teachers) (052)

EDRS PRICE MF01/PC06 Plus Postage.
DESCRIPTORS Behavioral Objectives; *Computer Literacy; Computer Oriented Programs; *Computers; *Computer Science Education; Course Content; Course Descriptions; Grade 9; Junior High Schools; Learning Activities; *Programing; State Curriculum Guides; *Word Processing
IDENTIFIERS Apple (Computer); Connecticut

ABSTRACT

This computer education curriculum is designed specifically for Connecticut's Regional Vocational Technical Schools' grade 9 computer education course. Each of the 24 lessons is expected to cover at least one class period of 50 minutes. Introductory materials include a listing of course goals and objectives, an outline of sequence and scope via learning outcomes, and recommended resource and reference materials. The components of each lesson may include some or all of the following: title; goal(s); learning outcomes; background information; lists of reference materials and materials needed; vocabulary list; lesson description (outline); hands-on, independent, other, and/or followup activities; assignment/activity sheet(s); study guide; and information sheets (handouts). Lessons cover introducing the Apple, components of the computer, input and output devices, history of computerization, computer systems, storage devices, introduction to software, keyboarding skills, computer applications, business applications, word processing, trade applications, uses and abuses of computers, computers and careers, programming logic, flowcharting, introduction to hands-on programming, and programming in BASIC. A glossary is appended. (YLB)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED264406

COMPUTER EDUCATION CURRICULUM

CONNECTICUT

VOCATIONAL TECHNICAL SCHOOL SYSTEM

Version 4: March 1985

Developed Under the Direction of

Dr. Kenneth Gray
Superintendent, Vocational Technical School System

U.S. DEPARTMENT OF EDUCATION
NATIONAL INSTITUTE OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

☐ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official NIE position or policy.

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

V Richards

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

Prepared by

Linda Kittell
J.M. Wright Regional Vocational Technical School

Joseph Walczak
Wilcox Regional Vocational Technical School

CE043087

BEST COPY AVAILABLE

Table of Contents

Overview..... 1

Course Goals and Objectives..... 2

Sequence and Scope..... 4

Recommended Resource and Reference Materials..... 9

Lesson 1: Introducing the APPLE..... 15

Lessons 2 and 3: Components of the Computer..... 17

Lesson 4: Input and Output Devices..... 23

Lesson 5: History of Computerization..... 26

Lesson 6: Computer Systems..... 29

Lesson 7: Storage Devices..... 31

Lesson 8: Introduction to Software..... 33

Lesson 9: Keyboarding Skills..... 36

Lesson 10: Computer Applications..... 38

Lesson 11: Business Applications..... 40

Lessons 12 and 13: Word Processing..... 43

Lesson 14: Word Processing Project..... 52

Lesson 15: Trade Applications..... 65

Lesson 16: Uses and Abuses of Computers..... 68

Lesson 17: Computers and Careers..... 71

Lesson 18: Programming Logic..... 73

Lesson 19: Flowcharting..... 77

Lesson 20: Introduction to Hands-on Programming..... 85

Lesson 21: Programming in BASIC - 2: PRINT, END, GOTO, DOS..... 91

Lesson 22: Programming in BASIC - 3: Graphics..... 97

Lesson 23: Programming in BASIC - 4: LET, READ-DATA, INPUT.....106

Lesson 24: Programming in BASIC - 5: IF-THEN, FOR-NEXT.....112

Glossary.....116

OVERVIEW

This computer education curriculum has been designed specifically for Connecticut's Regional Vocational Technical Schools' grade nine computer education course. The curriculum has been created with the following assumptions:

1. Each school may have a different number of class periods for their course. Twenty-four lessons (the minimum required number of class meetings) are outlined in detail. With the many hands-on and independent activities, the curriculum could be expanded greatly.

2. Each lesson is expected to cover at least one class period of 50 minutes.

3. Technical knowledge of the material included is not assumed. Teachers should be able to adequately prepare for each class by carefully reviewing the reference materials listed for that lesson and previous related lessons.

4. Although the curriculum has been put together as a detailed outline and structure for the course, the lessons have been created with flexibility to allow each teacher to focus on activities or concepts of their own choosing while at the same time covering each topic thoroughly.

5. Vocabulary important to a lesson is listed within each lesson. These terms are defined either in the glossary or within the lesson's materials that all teachers will have.

6. Reference materials listed for each lesson include materials that may not be available in each school. However, careful attention has been given to include only those materials that all schools will have in the "materials needed" sections.

COURSE GOALS AND OBJECTIVES

The Goals and Objectives that follow cover the basic information that is included in the curriculum. However, the order presented here does not reflect the actual sequence of lessons.

I. Background Information : The student will understand the basic concepts that underlie the development of computers in society.

A. Vocabulary : Students will be conversant with fundamental terms related to computer technology.

Lessons: detailed in all lessons

B. Components : Students will identify and understand the function of the major components of computer systems.

Lessons: 1, 2, 3, 4, 7, 8, 9

C. History : Students will appreciate the historical development of computerization.

Lessons: 5, 6

D. Ethics : Students will understand the importance of behaving in an ethical and responsible manner in relation to computers and information systems in society.

Lessons: 1, 16

II. Applications : The students will understand how they, as members of society can and will use computers.

A. Problem-solving Tools : Students will value and use computers as general-purpose machines designed, built and operated by humans as problem-solving tools.

Lessons: 8, 14, 18, 19

B. General Uses : Students will become aware of the variety of computer applications, their advantages, disadvantages, limitations and impact on individuals and organizations.

Lessons: 5, 6, 10, 11, 15, 16, 17

C. Impact on Jobs : Students will realize the impact of computerization on the job market as well as specific trade areas.

Lessons: 11, 15, 16 ,17

III. Computer Operation : Students will develop and appreciate a variety of techniques for exercising control over computers.

A. Basic Computer Operation : Students will become self-sufficient in the basic operations of the keyboard, peripheral devices and software manipulation.

Lessons: 1, 8, 9, 10, 11, 12, 13, 14

B. Programming Logic : Students will understand and apply simple logic, principles and techniques related to computer programming.

Lessons: 18, 19, 20, 21, 22, 23, 24

SEQUENCE AND SCOPE VIA LEARNING OUTCOMES

Lesson 1...Introducing the Apple

Students will
1. be able to name the major components of the Apple computer, know their locations and describe their functions
2. handle disks and components with care and respect
3. start the system from a cold start using the "System Master" disk
4. catalog and run demonstration programs on the "Sample Programs" disk

Lesson 2 & 3...Components of the Computer

Students will
1. recognize how data is stored in main computer storage
2. describe how the CPU functions in general terms
3. become familiar with how computer memory is measured
4. be aware of how the main storage areas differ (i.e., ROM vs. RAM)
5. be able to identify and define "hardware"

Lesson 4...Input and Output Devices

Students will
1. be able to name various input and output devices
2. describe the applications and utility of a variety of input and output devices
3. recognize specialized input and output devices found in the shop areas

Lesson 5...History of Computerization

Students will
1. understand the development of computer hardware which led to modern computing devices
2. understand the role played by computer software in the history of the computing industry
3. appreciate the important contributions of key figures in the computing industry

Lesson 6...Computer Systems

Students will
1. understand the input/process/output cycle
2. develop an awareness of the different sizes of machines available (mini, micro, mainframe)
3. appreciate the data storage, retrieval, inquiry, update, and sorting capabilities of the different computer systems

Lesson 7...Storage Devices

Students will
1. understand the need for auxiliary storage
2. be able to describe when the sequential and random access methods of storage are used
3. become familiar with storage devices and media used on large and small computers

Lesson 8...Introduction to Software

Students will
1. understand the difference between systems software and applications software
2. develop an appreciation of the use of symbolic code languages over binary codes
3. understand how hardware and software interact to make the computer a functional, useful tool

Lesson 9...Keyboarding Skills

Students will
1. learn to enter information to the computer efficiently through the Apple IIe keyboard
2. learn the home-row placement of the fingers to access the standard typewriter keys
3. become familiar with the use of special keys for entering data
4. learn the function of the special keys: return; left, right, up, and down arrows; shift; caps lock; control; open and closed apple; escape; delete; tab

Lesson 10...Computer Applications

Students will
1. practice using a menu driven program
2. become familiar with the different types of software written for classroom use
3. develop an awareness of business applications for the computer

Lesson 11...Business Applications

Students will
1. see and appreciate how an automated office uses both word and data processing
2. run and test some sample business oriented programs
3. use sample business accounting systems
4. be aware of how a Data Base Management System (DBMS) is used in a business environment
5. be shown how the marriage of a DBMS and word processing can help run a business

Lesson 12 and 13...Word Processing

Students will

1. be able to efficiently use a word processing program to enter, edit, and print text
2. become proficient in the following features of this program: a) Erase b) Unerase c) Move d) Moveback e) Find f) Replace g) Retrieve h) Save i) Init j) Delete k) Rename l) Print-Draft m) Print-Final n) Clear
3. practice these features by entering text supplied by instructor and making a series of changes necessary to bring the text to a desired format
4. print the reformatted text for a grade

Lesson 14...Word Processing Project

Students will

1. efficiently use a word processing program to complete an assignment or term paper
2. edit and format their paper as well as print two copies of a final draft. One copy will be handed in to the instructor for whom it was due and one copy for this lesson

Lesson 15...Trade Applications

Students will

1. develop an understanding of the significance computers will play in many if not all of the shop areas
2. be able to name several computer applications in the shop areas

Lesson 16...Uses and Abuses of Computers

Students will

1. be aware of the role computers in medicine, government, business, farming, education, manufacturing and design
2. be aware that half the U.S. economy is concerned with information processing
3. understand that future changes in information handling will give rise to electronic mail, electronic publishing and the electronic cottage
4. be aware of the dangers from invasion of privacy, computer theft, computer fraud and job displacement

Lesson 17...Computers and Careers

Students will

1. be able to identify main categories of computer related careers, ranging from the design and manufacture through application and operation
2. be able to describe specific jobs in each area, the nature of the work itself, and employment and earning potential
3. be able to outline the necessary education and job training as well as qualifications needed with each position

Lesson 18...Programming Logic

Students will

1. identify four computer languages: BASIC, FORTRAN, COBOL, and Pascal
2. identify the major area of use for each language
3. be able to list the five logical steps for solving a problem on the computer
4. be able to develop algorithms for the solution of non-computer problems

Lesson 19...Flowcharting

Students will

1. identify the four major flowchart symbols
2. predict the outcome of a simple flowcharted program
3. break a simple program into logical sections and make a flowchart

Lesson 20...Introduction to Hands-on Programming

Students will

1. understand that the initializing process writes DOS on the disk, establishes a catalog area and marks the sectors where programs will be stored
2. know how to initialize a blank disk using the System Master
3. understand the meaning of the following commands: NEW, LIST, RUN, SAVE, INIT, CATALOG
4. know the BASIC symbols for addition, subtraction, multiplication, division, and raising a value to a power
5. know the mathematical order of operations
6. be aware of computer detected errors

Lesson 21...Programming, using PRINT, END, GOTO, editing, Disk Operating System

Students will

1. understand the main parts of a BASIC statement are the line number, keyword, and body
2. understand the meaning of the keywords PRINT, END, GOTO
3. learn how to correct, add, or delete a program line
4. use the GOTO statement to control program execution and CONTROL-C to stop an endless loop
5. understand the concept of the loop
6. understand the meaning of the following commands: LOAD, DELETE, LOCK, UNLOCK, RENAME

Lesson 22...Graphics

Students will

1. experience using low and high resolution graphics statements to display pictures on the screen
2. know the meaning of BASIC low resolution keywords: GR, COLOR=, PLOT, HLIN, VLIN
3. know the meaning of BASIC high resolution keywords: HGR, HGR2, HCOLOR=, HPLOT
4. understand simple graphics capabilities of the computer

Lesson 23...Programming using LET, READ...DATA, INPUT (for the more interested students)

Students will

1. know the difference between a variable and a constant
2. understand that numeric variables store numbers
3. know that variables may be single letters, double letters, or single letters followed by single numbers
4. understand that the LET and READ...DATA statements assign values to variables from within the program
5. understand that the INPUT statement allows the user to assign values to variables
6. understand how to use string variables
7. experience the use of variables in prepared programs

Lesson 24...Programming using IF...THEN and FOR...NEXT (for advanced students)

Students will

1. know the functions of the comparison symbols: <, <=, >, >=, =, <>
2. know how to use the IF...THEN statement to PRINT, GOTO, LET, or END, based on a condition being true or false
3. use the FOR...NEXT statements to execute a set of instructions a given number of times
4. understand the distinction between an endless and a controlled loop

RECOMMENDED RESOURCE AND REFERENCE MATERIALS

Title	Publisher	Description
"Airline reservation Simulation" Program	Mark Crane, Abbott RVTS	Software program that demonstrates real-time vs. batch processing. (Intro to Computer Disk - Ch. 8)
"Apple Presents Apple"	Apple	Software package introducing the functions of various parts of the computer, especially the keyboard
Apple II DOS Manual	Apple	Manual packaged with disk drive. Describes use of the DOS and serves as a reference guide to DOS
Apple IIe Owner's Manual	Apple	Manual packaged with all IIe computers - the "Bible"
Applesoft Tutorial For IIe Only	Apple	The complete manual for learning to program the IIe computer
Applesoft BASIC Programmer's Reference Manual	Apple	Instructions for advanced Apple IIe programming
"Bank Street Writer"	Broderbund	Instructional word processing software
Basic Basic	Dilithium Press	Resource material for teaching programming
Basic Computer Programming for Kids	Prentice Hall	Resource materials for teaching programming

Title	Publisher	Description
Beginning Applesoft BASIC	MECC	A teacher training program containing many activities and classroom materials for an introduction to BASIC
"Careers in Computer Science and Service"	Center for the Humanities	Slide/tape program that looks at all kinds of job opportunities in hardware, software and operations. Covers required education, training, experience and opportunities
"Computer Hardware - What it is and How it Works"	Center for the Humanities	Slide/tape program that examines computer operations, describes typical micro-computers, and presents actual processing of information
Computer Literacy: Problem Solving with Computers by Carin Horn and James Poirot	Sterling Swift 1981	Comprehensive text covering all facets of computer literacy
Computer Literacy: An introductory Course	The Continental Press 1982	A 15 lesson course with accompanying software transparancies and excellent activities
Computer Literacy Show and Tell Kit	Sterling Swift	Notebook containing various computer components
"Computer Software - What it is and How it Works"	Center for Humanities	Slide/tape program that shows how software enables a computer to process information
"Computer: Peripherals to Programming"	RMI Media Productions, Inc.	Media presentation covering memory, storage, I/O devices and programming languages

Title	Publisher	Description
Are you Computer Literate by Karen Billings and David Morsund	Dilithium Press	Background info, articles, quizzes for computer literacy courses
"From Pebbles to Programs"	Center for the Humanities	Filmstrip/cassette program describing the history of computerization, key terminology and a broad variety of computer applications
Fundamentals of Digital Computers by Donald Spencer	Howard Sams	Reference book on history, functions and applications of computers including an extensive glossary
"History" program	Mark Crane Abbott RVTS	Software program which presents the history of computerization in drill and practice format (Intro to computers Disk - ch 2)
I Speak BASIC to My Apple	Hayden	Programming reference material with many classroom activities appropriate for overheads
"Introduction to Computers: the CPU"	RMI	Media presentation which covers the development and uses of computers plus the CPU and major components
Know Your Apple IIe	Muse	Software package using graphics to introduce the parts and functions of the computer
"Making it Count" videotape	CPTV	A complete introduction to computers that was presented on CPTV. Videotapes are available from Charles White

Title	Publisher	Description
Master Type	Lightening Software	Software package for learning the keyboard and improving typing skills in game format
Meet the Computer	Sunburst	Color coded activity folders that teach programming statements and commands
Nailing Jelly to a Tree by Willis and Danley	Dilithium Press	Reference manual for BASIC programming with many hands-on activities
One Computer 30 Kids	MEKA Publishing	Workbook of programming activities that do not require sitting at the computer
"Personal Computers: Basic Concepts"	RMI	Media presentation which introduces the microcomputer and its uses in the house, school, and business
The Secret Guide to Computers Volumes 1 through 8 by Russ Walter	Russ Walter	"Fun" approach through hands on-experience to programming, systems and languages
Hi-Res Architectural Design	Avante-Guarde Creations	Software program that permits designing of floor plans (Requires hand controllers)
Apple Writer IIe	Apple	Word processing Software
Apple IIe User's Guide (Lon Poole)	Osborne/McGraw-Hill (second edition)	Reference material on the use and programming of the Apple
A Guide to Programming in Applesoft (Bruce Presley)	Lawrenceville Press (second edition)	Resource material for teaching programming

Title	Publisher	Description
Apple IIe User's Handbook (Weber Systems Staff)	Weber Systems	Reference material on the use and programming of the Apple
The Elementary Apple (William B. Sanders)	Datamost	Reference material on the use and programming of the Apple
Kids and the Apple (Edward H. Carlson)	Datamost	Resource material for teaching programming (designed for about grade 7)
Bank Street Speller	Broderbund	Software used in conjuction with word processing software to locate and correct spelling errors
Apple Basic (Richard Haskell)	Prentice Hall	Reference material for graphics
Practicing Programming on the Apple II/IIe (Maryann Marrapodi, et. al.)	Random House	Resource material for teaching programming
Spotlight on Computer Literacy by Ellen Richman	Random House 1982	Student and teacher materials for a 24 lesson course. Excellent reference material.
701 - Apple Demonstration	MECC	Applications of drill and practice, tutorial, simulation problem solving and work sheet generation are demonstrated
704 - Social Studies Volume 3	MECC	Includes programs to simulate business management and elementary economic principles
706 - Math Volume I	MECC	Programs to teach logic while reinforcing plotting points and angle measurements

Title	Publisher	Description
713 - Business Volume II	MECC	Program simulating a total payroll system. Requires dual disk drives and a 132 character printer
714 - Business Volume I	MECC	Programs in banking, business management and monetary principles are included
721 - Business Volume III	MECC	Package provides a realistic experience with automated accounting systems
Minute Manual for Apple Writer IIe by Jim Pirisino	MinuteWare	Reference material for easy use of Apple Writer IIe

LESSON NUMBER 1

TITLE Introducing the Apple

GOALS

Components : Students will identify and understand the function of the major components of computer systems.

Basic Computer Operation : Students will become self-sufficient in the basic operations of the keyboard, peripheral devices and software manipulation.

Ethics : Students will understand the importance of behaving in an ethical and responsible manner in relation to computers and information systems in society.

LEARNING OUTCOMES

1. Students will be able to name the major components of the Apple computer, know their locations and describe their functions.
2. Students will handle disks and components with care and respect.
3. Students will start the system from a cold start using the "System Master" disk
4. Students will catalog and run demonstration programs on the "Sample Programs" disk.

BACKGROUND INFORMATION

Similarities and differences between the standard typewriter keyboard and the computer keyboard should be carefully noted. Analogies can be formed showing the commonalities between audio storage on a standard cassette tape and the storage of data on a disk.

REFERENCE MATERIALS

1. Spotlight on Computer Literacy, pages 5 to 11
2. Computer Literacy: Problem Solving with Computers, pages 167 to 185
3. "Apple Presents Apple" software package
4. Apple IIe Owner's Manual, pages 11 to 41

MATERIALS NEEDED

1. "System Master" disk
2. "Know Your Apple" disk
3. "Sample Programs" disk

VOCABULARY

Power supply, AC current, DC current, CPU, RAM, ROM, integrated circuit, CATALOG, RUN, menu, computer, monitor, disk drive, printer

LESSON DESCRIPTION

I. Open computer case, identify major components

II. Describe disk functions

   A. Show proper disk handling techniques

   B. Show proper insertion into disk drive

III. Demonstrate proper computer start-up using "System Master" disk and "Sample Programs" disk

   A. Demonstrate "CATALOG" using both disks
   B. Run sample programs

HANDS-ON ACTIVITIES

1. Using the "Sample Programs" disk, students should run "Applevision", "Color Test" as well as other programs of their choosing.
2. Students should run "Apple Presents Apple" to become comfortable with handling disks and using the keyboard.

LESSON NUMBER 2 and 3

TITLE Components of the Computer

GOAL Components : Students will identify and understand the functions of the major components of computer systems.

LEARNING OUTCOMES

1. Students will recognize how data is stored in main computer storage.
2. Students will describe how the CPU functions in general terms.
3. Students will become familiar with how computer memory is measured.
4. Students will be aware of how the main storage areas differ (i.e. ROM vs. RAM).
5. Students will be able to identify and define "Hardware".

BACKGROUND INFORMATION Carefully preview the "Computer Hardware" slide/tape presentation. Study guide may be used when viewing.

REFERENCE MATERIALS

1. Spotlight on Computer Literacy, pages 20-36
2. Computer Literacy: Problem Solving with Computers, pages 167-185
3. Apple IIe Owner's Manual, pages 45-53
4. "Introduction to Computers: the CPU" filmstrip program

MATERIALS NEEDED

1. "Computer Hardware" slide/tape presentation
2. "Know your Apple" disk

VOCABULARY — RAM, ROM, CPU, ALU, control, byte, bit, binary, buffer, address bus, data bus, I/O, peripheral, interface, hardware

LESSON DESCRIPTION

I. Show slide and tape presentation "Computer Hardware"

A. CPU is the 'Brains' of the computer

B. CPU is composed of control and the ALU

C. CPU understands only binary

D. Control keeps track of data movement

E. ALU does all math and logic operations

F. ROM is the equivalent of a permanent memory

G. RAM is a temporary or erasable memory

1. Each memory location is composed of 8 bits (switches)

a. 8 bits = 1 byte

b. Approximately 1000 bytes = 1 kilobyte = 1 K

2. Each memory location can hold one character (number or letter)

3. Each memory location has a unique number called address

H. The CPU consults with ROM to interpret what to do with the data in RAM

I. The wires or electrical pathways that connect RAM, CPU, ROM and I/O devices are called busses

HANDS ON ACTIVITIES

1. Remove cover from the computer and identify as many components as possible. Use Apple IIe Owner's Manual for reference.

2. All students should run all four sequences on the "Know Your Apple" disk. Students may use assignment sheet as a guide.

INDEPENDENT ACTIVITIES

Students should complete Assignment 2/3, Components Worksheet.

OTHER ACTIVITIES

1. Obtain copies of different computer magazines. Make a guess as to who reads them and for what purposes. Some magazines are concerned with "hardware"; others with "software". CREATE an effective advertisement for a particular piece of computer hardware, such as a CRT, printer, computer, or disk drive. Explain why a consumer should buy this product over any other.

2. Get copies of computer hardware catalogs (brochures are also acceptable). Study the specifications of different computers offered for sale. Discuss differences between models and brands, and criteria for deciding which computer to buy.

3. Many people are afraid that the use of computers will lead to unemployment. Collect information and be prepared to discuss this concern.

4. Write a one-sentence message in ASCII. Exchange messages and translate them back into English. The following is the ASCII representation of the letters of the alphabet, numbers, and punctuation:

01000001	A	01001110	N	00110000	0	00110111	7
01000010	B	01001111	O	00110001	1	00111000	8
01000011	C	01010000	P	00110010	2	00111001	9
01000100	D	01010001	Q	00110011	3	00101100	,
01000101	E	01010010	R	00110100	4	00101101	.
01000110	F	01010011	S	00110101	5	00111111	?
01000111	G	01010100	T	00110110	6	00100000	(SPACE)
01001001	I	01010110	V				
01001010	J	01010111	W				
01001011	K	01011000	X				
01001100	L	01011001	Y				
01001101	M	01011010	Z				

ASSIGNMENT SHEET FOR MUSE "KNOW YOUR APPLE IIe"

Answer the following questions as you view the program. Be brief, but complete.

1. What is the monitor? What does it do?
2. What is a prompt? Show the symbols and names for three prompts.
3. What is a cursor? What is it used for?
4. What three formats are used to present informaton on the monitor? Explain and describe each.
5. What is the disk drive? What is its purpose?
6. What is the disk? What is it used for?
7. Explain the purpose of the write enable notch.
8. What are other names for the disk? Why are these names used?
9. Why should a disk never be bent? Explain the precautions in handling disks.
10. How long do disks last? Explain.
11. How many tracks and sectors does a disk have? Explain.
12. What are the disk capacities?
13. What does DOS stand for?
14. What does "boot" (or booting) mean?
15. What does DOS do? Explain.
16. What is a "catalog"?
17. List the steps in using the disk drive.
18. What check(s) should be made before inserting or removng a disk in drive?
19. Where is the power supply located?
20. How is the power cord connected? Explain.
21. What happens when the power switch is turned on?
22. How is the video cable connected? Explain.
23. Can a TV be connected to the VIDEO OUT? Explain.
24. What does the RF Modulator do? How is it connected?
25. What "things" can be connected to the Apple? Name them.
26. What is the keyboard used for? Explain.
27. How many keys does it have?
28. What does ASCII stand for?
29. How is a letter or symbol repeated with the IIe?
30. What is the SHIFT key used for?
31. What is the CAPS LOCK key used for?
32. What does the CONTROL key do?
33. What is the function of the TAB key?
34. What are the arrow keys used for?
35. What is the purpose of the RETURN key?
36. What do the open & closed apple keys do?
37. What does the RESET key do?
38. What is the "motherboard"?
39. What is the 6502? What does it do? What does CPU stand for?
40. What does ROM stand for? What does it mean?
41. What does RAM stand for? What does it mean?
42. Where does ROM data come from? Where does RAM data come from?
43. What do interface cards do?
44. What is the standard slot for disk drive card? For printer card? For 80-column card?
45. What does the power supply do?
46. What does the LED tell you?

STUDY GUIDE (Computer Hardware - What it is and How it Works)

ANSWER THE FOLLOWING QUESTIONS. BE BRIEF. BE COMPLETE. WRITE CLEARLY.

=====PART ONE=====

1. What is input? output? process?
2. What is memory (or internal memory)? What does it do?
3. What does CPU stand for? What are the two main parts? What functions do they perform?
4. What does ALU stand for?
5. What does CRT stand for? What is its purpose?
6. What is a binary device?

=====PART TWO=====

1. What is the most widely used input device? output device?
2. What is a terminal?
3. What is a disadvantage of just having a CRT as an output device? How can this be overcome?
4. What is an input device? output device? information storage system?

=====PART THREE=====

1. What is permanent memory? temporary memory? Explain.
2. What does "writing information into memory mean? What does "reading information from memory" mean?
3. What is the temporary memory of a computer called? What is the permanent memory called?
4. What does RAM stand for? What does ROM stand for? What is stored in each?
5. What is a "bit? a "byte"? How many bits are in a byte?
6. What does "K" represent?

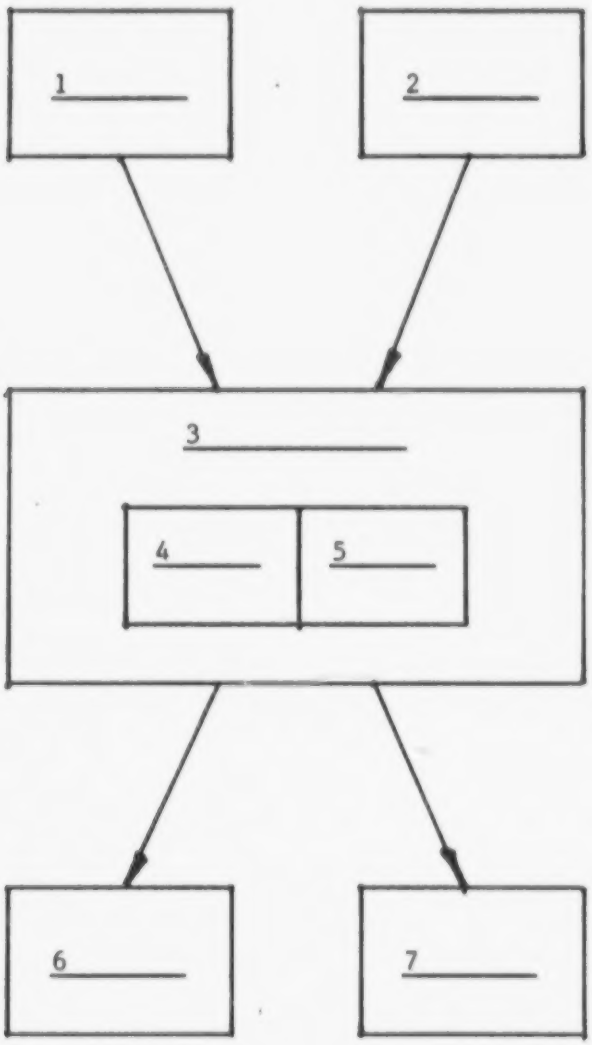
=====PART FOUR=====

1. What two tools does the control (or control unit) have? What does each do?
2. What is a program?
3. What is a "bus"? Identify two kinds of busses.

ASSIGNMENT 2/3

COMPONENTS

Place the proper component name from the list on the right in the numbered spaces on the diagram.



ALU

CPU

Control

Keyboard

Peripheral Storage

Printer

Video screen

LESSON NUMBER 4

TITLE Input and Output Devices

GOAL Components : Students will identify and understand the function of the major components of computer systems.

LEARNING OUTCOMES

1. Students will be able to name various input and output devices.
2. Students will describe the applications and utility of a variety of input and output devices.
3. Students will recognize specialized input and output devices found in the shop areas.

BACKGROUND INFORMATION In order to fully understand how a computer processes information, students should be familiar with the wide variety of mechanisms whereby information "gets into" and "spits out" of the main kinds of computers.

REFERENCE MATERIALS

1. Spotlight on Computer Literacy , pages 5-19
2. Apple IIe Owner's Manual , pages 95-101
3. Computer Literacy: Problem Solving with Computers , pages 167-185
4. "Computers: Peripherals to Programming" filmstrip program

MATERIALS NEEDED

1. Computer Literacy Show and Tell Kit
2. "Know Your Apple" disk
3. Input and output devices (see lists in the lesson description)

VOCABULARY — Parallel interface, series interface, OMR, card reader, scanner, dot matrix printer, digital, analog, CRT, VDT, input, output, MICR

LESSON DESCRIPTION

I. Discuss and demonstrate wherever possible the functions of input devices.

   A. Apple keyboard

   B. tape recorder and tape

   C. disk drive and disk

   D. hard disk drive

   E. tape drives

   F. card reader and punched cards such as the Hollorith code card in Show and Tell Kit

   G. light pen

   H. keypunch machine

   I. MICR (Magnetic Ink Character Recognition) reader and a cancelled check to show the special "computer characters"

   J. optical mark reader such as a Scan-tron unit and sample test answer forms

II. Discuss and demonstrate wherever possible the functions of output devices.

   A. Apple monitor

   B. television set

   C. CRT or VDT

   D. printer

   E. plotter

   F. terminal

III. Discuss the difference between digital and analog information.

IV. Discuss and demonstrate input and output devices in the shop areas in your school.

HANDS-ON ACTIVITIES

Open top of Apple and determine where the I/O devices are attached, if interface is parallel or series, and which interfaces are built into the motherboard.

INDEPENDENT ACTIVITIES

1. Have each student research and write a paragraph describing a peripheral device that would have an application in one of the shop areas in the school.

2. Students may need to review parts of the "Know Your Apple" disk.

3. Do Spotlight on Computer Literacy activities on pages 16 and 19.

LESSON NUMBER 5

TITLE History of Computerization

GOAL

History : Students will appreciate the historical development of computerization.

General Uses : Students will become aware of the variety of computer applications, their advantages, disadvantages, limitations and impact on individuals and organizations.

LEARNING OUTCOMES

1. Students will understand the development of computer hardware which led to modern computing devices.
2. Students will understand the role played by computer software in the history of the computing industry.
3. Students will appreciate the important contributions of key figures in the computing industry.

BACKGROUND INFORMATION

An introduction to the development of the computer throughout history is critical to maintaining a proper perspective of where we are and where we are going with computerization in our world today.

REFERENCE MATERIALS

1. Spotlight on Computer Literacy , pages 37 to 50
2. Are You Computer Literate , pages 27 to 38
3. Computer Literacy: Problem Solving with Computers , pages 27 to 43
4. "From Pebbles to Programs", Center for the Humanities, slide and tape presentation

MATERIALS NEEDED

1. "Making it Count" videotape
2. "History" disk

VOCABULARY

abacus, Jacquard's loom, Babbage's analytical engine, Hollerith code, electromechanical computer, digital computer, ENIAC, UNIVAC, vacuum tube, transistor, integrated circuit, chip

LESSON DESCRIPTION

I. Discuss the following developments in the history of the need to compute
   - A. abacus
   - B. arithmetic machines
   - C. Jacquard's loom
   - D. Babbage's analytical machine
   - E. Hollerith's tabulating machine

II. Compare and contrast the four generations of electronic computers including the following
   - A. speed of calculation
   - B. vacuum tubes
   - C. transistors
   - D. integrated circuit chips
   - F. size differences

HANDS-ON ACTIVITIES

Have all students run the "History" disk.

INDEPENDENT ACTIVITIES

Have students research the "Hollerith code" and map what a punched card of their names would look like.

Have students work on mini-research project.

FOLLOWUP ACTIVITIES

View the "Making it Count" videotape.

SHORT RESEARCH PROJECT

TURN IN THE ANSWERS TO THE FOLLOWING. THIS IS A MINI RESEARCH PROJECT. USE THE SCHOOL OR PUBLIC LIBRARY, ENCYCLOPEDIA, REFERENCE BOOKS, ETC.

LAST NAME A THROUGH M -

1. What was UNIVAC?
2. Who was John Napier? What did he invent that may have influenced computers?
3. Who was Herman Hollerith? What did he do?
4. When, and why, were transistors developed?
5. When did the age of "modern computers" begin?

LAST NAME N THROUGH Z -

1. What was ENIAC?
2. Who was Blaise Pascal? What did he invent that may have influenced computers?
3. Who was Charles Babbage? What did he do?
4. When, and why, were integrated circuits developed?
5. What is the abacus? When was it developed?

WRITE OUT YOUR COMPLETE ANSWER. AFTER EACH ANSWER, INDICATE YOUR SOURCE OF INFORMATION FOR THE ANSWER.

LESSON NUMBER 6

TITLE Computer Systems

GOALS

History : Students will appreciate the historical development of computerization.

General Uses : Students will become aware of the variety of computer applications, their advantages, disadvantages, limitations and impact on individuals and organizations.

LEARNING OUTCOMES

1. Students will understand the input/process/output cycle.
2. Students will develop an awareness of the different sizes of machines available (mini, micro, mainframe).
3. Students will appreciate the data storage, retrieval, inquiry, update, and sorting capabilities of the different computer systems.

BACKGROUND INFORMATION

A computer system consists of the electronic computer along with peripheral devices which function as one unit. The size, speed, cost and applications of a computer system are determined by its component parts.

REFERENCE MATERIALS

1. Computer Literacy: Problem Solving with Computers, pages 145 to 166
2. Spotlight on Computer Literacy, pages 51 to 68

MATERIALS NEEDED

1. "Airline Reservation Simulation" disk
2. "Making it Count: videotape

VOCABULARY

time-sharing, microcomputer, minicomputer, mainframe, networking, telecommunication, database, retrieval, modem, acustic coupler, batch processing, realtime processing

LESSON DESCRIPTION

I. Using airline reservations as an example of realtime processing, discuss the process of making a reservation for a particular flight by phone. Include the following concepts

   A. large database

   B. telecommunication

   C. data retrieval

   D. inquiry, update

   E. random nature of system

II. Discuss the process of state payroll checks as done by computer as an example of batch processing. Include the following concepts

   A. timing

   B. sequential nature of payroll

   C. use of computer cards

   D. repetitive nature of payroll

   E. simplicity of program (rate * time = pay)

III. Discuss the attributes and applications of micro, mini and mainframe systems. Elicit various functions that computers perform and determine the most efficient and cost effective system with which to perform each task.

HANDS-ON ACTIVITIES

Have each student run the "Airline Reservation Simulation" disk.

INDEPENDENT ACTIVITIES

Students may tour area business and shops within the school with mainfame or mini computer systems.

LESSON NUMBER 7

TITLE Storage Devices

GOAL Components : Students will identify and understand the function of the major components of computer systems.

LEARNING OUTCOMES

1. Students will understand the need for auxillary storage.
2. Students will be able to describe when the sequential and random access methods of storage are used.
3. Students will become familiar with storage devices and media used on large and small computers.

BACKGROUND INFORMATION Familiarization with the mechanisms for storing and retrieving data via primary and secondary storage devices is important to the complete understanding of a total computer system.

REFERENCE MATERIALS

1. Computer Literacy: Problem Solving with Computers , pages 167 to 186
2. Apple IIe Owner's Manual , pages 57 to 69
3. Apple IIe DOS Manual
4. Spotlight on Computer Literacy , pages 1 to 7
5. "Computers: Peripherals to Programming" filmstrip program
6. "Computer Hardware" slide and tape program, part 3

MATERIALS NEEDED

1. Show and Tell Kit
2. Cassette player
3. Any disk based computer program that is also copied onto a cassette

VOCABULARY

random access, sequential access, record, file, field, disk, disk pack, floppy disk, magnetic tape, DOS, drum storage

LESSON DESCRIPTION

1. Again, discuss the examples of airline reservations vs. payroll to show the need for random access in the former while sequential (tape) storage would be adequate for the latter.
2. Demonstrate the mobility of the random access (disk drive) read/write heads in contrast to the stationary heads in sequential access (tape drive). A disk drive may be partly dismantled to show actual parts and/or use the "Show and Tell Kit" to show tape, disk, etc.
3. Compare information format on a computer card with that of tape and disk.
4. Compare attributes of programs on cassette versus disk such as speed of loading, program complexity, menu driven.

HANDS-ON ACTIVITIES

Connect a cassette tape player to at least one computer. Have students compare loading time and basic operation of the tape program in contrast to the disk version.

LESSON NUMBER 8

TITLE Introduction to Software

GOALS

Components : Students will identify and understand the function of the major components of computer systems.

Problem-solving Tools : Students will value and use computers as general-purpose machines designed, built and operated by humans as problem-solving tools.

Basic Computer Operation : Students will become self-sufficient in the basic operations of the keyboard, peripheral devices and software manipulation.

LEARNING OUTCOMES

1. Students will understand the difference between systems software and applications software.
2. Students will develop an appreciation of the use of symbolic code languages over binary codes.
3. Students will understand how hardware and software interact to make the computer a functional, useful tool.

BACKGROUND INFORMATION

Utilization of "canned" software is the primary method of software access in schools, homes and most businesses. Students must become both familiar and comfortable with handling software.

REFERENCE MATERIALS

1. Computer Literacy: Problem Solving with Computers , pages 167 to 226
2. Apple IIe Owner's Manual
3. Apple IIe DOS Manual

MATERIALS NEEDED

1. "Computer Software" slide and tape presentation
2. "Sample Programs" disk

VOCABULARY

systems programming, applications programming, BASIC, COBOL, operating system, monitor, DOS, assembler, compiler, interpreter

LESSON DESCRIPTION

I. Show slide presentation

II. Discuss the following concepts

   A. Systems Programming

   B. Applications programming

   C. Computer languages

   D. Operating systems

   E. Monitor

   F. DOS

   G. Assembler

   H. Compiler

   I. Interpreter

III. Discuss and demonstrate the Apple DOS, monitor and interpreter. (The different prompts indicate which is activated. Use the user's manual as a reference.)

HANDS-ON ACTIVITIES

Have students check what commands will and will not work if a disk is not booted. Try CATALOG, LOAD, SAVE. These must be DOS commands. Load Sample Programs disk. Notice if error messages differ between DOS commands and BASIC commands.

INDEPENDENT ACTIVITIES

Run the phone list program on Sample Programs disk.

STUDY GUIDE (Computer Software - What it is and How it Works)

ANSWER THE FOLLOWING QUESTIONS. BE BRIEF. BE COMPLETE.

======PART ONE======

1. What is a computer program?
2. What is data?
3. What is hardware? What is software?
4. What is machine language?
5. Why is a zero often shown with a slash through it?
6. What does ASCII stand for?
7. What is a high-level language?
8. What is an operating system?
9. What does DOS stand for?

======PART TWO======

1. What two functions does a line number perform?
2. Why is it common practice to number lines by tens?
3. In BASIC, what does the asterisk (*) stand for?
5. What is the buffer? What is its function?

======PART THREE======

1. What is an "unconditional branch"? a "conditional branch"?
2. What do the following commands mean?
   PRINT
   END
   INPUT

======PART FOUR======

1. What is BASIC? What does BASIC stand for?
2. What do the following commands mean?
   GOSUB
   RETURN

LESSON NUMBER 9

TITLE Keyboarding Skills

GOALS

Basic Computer Operation : Students will become self-sufficient in the basic operations of the keyboard, peripheral devices and software manipulation.

Components : Students will identify and understand the function of the major components of computer systems.

LEARNING OUTCOMES

1. Students will learn to enter information to the computer efficiently through the Apple IIe keyboard.
2. Students will learn the home-row placement of the fingers to access the standard typewriter keys.
3. Students will become familiar with the use of special keys for entering data.
4. Students will learn the function of the special keys: return; left, right, up and down arrows; shift; capslock; control; open and closed apple; escape; delete; tab.

BACKGROUND INFORMATION

It has been projected that "keyboarding" will replace "typing" both in instruction as well as virtually all applications in the very near future. The standard typewriter may soon become a museum piece. Skill in the use of the computer keyboard is, even now, required of a great many workers whose primary job responsibility has nothing to do with typing.

REFERENCE MATERIALS

1. "Know Your Apple" disk
2. Apple IIe Owner's Manual , pages 32 to 41
3. Secret Guide to Computers , Volume 1, pages 5 to 8

MATERIALS NEEDED

1. "Apple Presents Apple" disk
2. "Master Type" disk

VOCABULARY keyboarding

LESSON DESCRIPTION

1. Describe special (non-alphanumeric) keys on the keyboard, explain their functions.
2. Explain the "home row" placement of hands on the keyboard. Discuss the advantages of memorizing the location of the keys.
3. Students should run "Apple Presents Apple" disk using all of the given menu options.
4. Students should then use the "Master Type" program to practice their skills using standard keys.

INDEPENDENT ACTIVITIES Students should use a manual typewriter and electric typewriter to type a short assignment to compare the different types of keyboards.

LESSON NUMBER 10

TITLE Computer Applications

GOALS

Basic Computer Operation : Students will become self-sufficient in the basic operations of the keyboard, peripheral devices and software manipulation.

General Uses : Students will become aware of the variety of computer applications, their advantages, disadvantages, limitations and impact on individuals and organizations.

LEARNING OUTCOMES

1. Students will practice using a menu driven program.
2. Students will become familiar with the different types of software written for classroom use.
3. Students will develop an awareness of business application for the computer.

BACKGROUND INFORMATION

Not long ago, the only applications that computers had were in the business and scientific communities. But with the advent of the personal computer things began to change. The computer could not only manage business but also become a tireless teacher and entertainer.

REFERENCE MATERIALS

1. Apple IIe Owner's Manual : pages 73 to 83
2. Spotlight on Computer Literacy , pages 51 to 63
3. Computer Literacy: Problem Solving with Computers , pages 45 to 93
4. Are You Computer Literate? , pages 91 to 109
5. "Personal Computers: Basic Concepts" filmstrip program

MATERIALS NEEDED

1. "MECC Demonstration" disk number 701 and support booklet
2. "MECC Business-Volume I" disk number 714 and support booklet
3. "MECC Mathematics-Volume I" disk number 706 and support booklet
4. "Sample Programs" disk

VOCABULARY drill and practice, CAI, tutorial, simulation

LESSON DESCRIPTION

1. Describe to students how computers have evolved into great problem solving tools for business, education and the home.
2. Load and run "MECC Demonstration" disk number 701. Run through all eight menu options to illustrate a) Drill and Practice b) Tutorial c) Simulation d) Games e) Problem Solving.
3. Run "MECC Business-Volume I" disk number 714 to show some of the common business applications of the computer such as money management and payroll programs.

HANDS-ON ACTIVITIES Students should run disks 701 and 714.

INDEPENDENT ACTIVITIES

1. Students can run MECC disk number 706 for exposure to more drill and practice and simulation programs.
2. Students should run Applevision or other programs on the "Sample Programs" disk.

LESSON NUMBER 11

TITLE Business Applications

GOALS

General Uses : Students will become aware of the variety of computer applications, their advantages, disadvantages, limitations and impact on individuals and organizations.

Impact on Jobs : Students will realize the impact of computerization on the job market as well as specific trade areas.

Basic Computer Operation : Students will become self-sufficient in the basic operations of the keyboard, peripheral devices and software manipulation.

LEARNING OUTCOMES

1. Students will see and appreciate how an automated office uses both word and data processing.
2. Students will run and test some sample business oriented programs.
3. Students will use sample business accounting systems.
4. Students will be aware of how a Data Base Management System (DBMS) is used in a business environment.
5. Students will be shown how the marriage of DBMS and word processing systems can help run a business.

BACKGROUND INFORMATION

Enormous amounts of money have been jointly spent by private enterprise and governmental agencies to develop applications for computer technologies. The most widespread non-governmental use of computers is in business.

REFERENCE MATERIALS

1. Spotlight on Computer Literacy , pages 51 to 63
2. Computer Literacy: Problem Solving with Computers , pages 45 to 47

MATERIALS NEEDED

1. "MECC Social Studies-Volume 3" disk number 704 and support booklet
2. "MECC Business-Volume 2" disk number 713 and support booklet
3. "MECC Business-Volume 3" disk number 721 and support booklet
4. "Sample Programs" disk
5. Sample "Junk Mail"

VOCABULARY DBMS, spreadsheet, word processing

LESSON DESCRIPTION

1. Explain to students how a DBMS program like a mailing list can be incorporated into a word processing program to produce much of the "Junk Mail" type of material that arrives in most people's homes. This is the type of standard form letter into which the computer inserts, as it is printing, a name and address that is stored in its mailing list. Instructor should ask students to bring to class any type of mail such as this for class discussion.
2. Run MECC disk 704 and use the program "Sell Bicycles", "Sell Apples" and "Sell Lemonade" to demonstrate how a microcomputer can help in keeping the books of a company and keeping track of profits.
3. Run MECC disk 714 and use the Payroll, Taxes and Loanam programs to show how businesses use the computer to keep exact records on employees and money.

HANDS ON ACTIVITIES

1. Run "Phone List" program from the "Sample Programs" disk to illustrate a simple DBMS program. From Phone List, have students 1) Enter names 2) Delete names 3) Search a listing 4) Print a file.

2. Have students run MECC disk 721. This disk illustrates a Microcomputer Integrated Computerized Accounting System (MICA). It includes a) General Ledger b) Accounts Payable c) Accounts Receivable and d) Inventory Control. It requires dual disk drives and a 132 character printer.

LESSON NUMBER 12 and 13

TITLE Word Processing

GOAL Basic Computer Operation : Students will become self-sufficient in the basic operations of the keyboard, peripheral devices and software manipulation.

LEARNING OUTCOMES

1. Students will be able to efficiently use a word processing program to enter, edit, and print text.

2. Students will become proficient in the following features of this program: a) Erase b) Unerase c) Move d) Moveback e) Find f) Replace g) Retrieve h) Save i) Init j) Delete k) Rename l) Print-Draft m) Print-Final n) Clear.

3. Students will then practice these features by entering text supplied by instructor and making a series of changes necessary to bring the text to a desired format.

4. Students will then print the reformatted text for a grade.

BACKGROUND INFORMATION One of the most important business applications in today's business world is word processing. This is also one of few programs that students can use for real, personally meaningful work - not just simulations.

REFERENCE MATERIALS "Bank Street Writer" support booklet

MATERIALS NEEDED

1. "Bank Street Writer" program disk

2. One blank disk per pupil

VOCABULARY editing, formatting, block movement, menus, margins, page numbering, justification

LESSON DESCRIPTION

1. Load "Bank Street Writer"on each of the computers. Then insert a blank disk in each of the drives.

2. Each copy of this program has a Tutorial on the reverse side. If time allows, have the students run through all of menu options.

3. If the tutorial is not used, then the instructor should personally go through the major features on the menu especially a) Erase b) Move c) Find d) Replace.

4. The "Transfer Menu" option should then be used to give the additional functions of a) Retrieve b) Save c) Print-Draft and d) Print-Final.

5. With each of these options, the students should enter any available text and then practice using the word processor to make changes or deletions as is necessary to become proficient.

6. Once the students become comfortable with editing and formatting, they should try printing their text in either a draft or final form. At this stage they should be shown the several print options such as margin settings, page numbering, etc.

7. Students should then practice saving and retrieving their text files using the blank disks provided.

8. When all functions have been used and reviewed, use the attached Word Processing Test to determine if the student has made sufficient progress in this lesson.

HANDS-ON ACTIVITIES Have all students complete assignment 12/13, Word Processing.

INDEPENDENT ACTIVITIES

Students may wish to use another version of a word processor such as "Apple Writer" and compare the features found on both. With knowledge of other word processing programs, the student could then write a short paper on their subject using the program that, in their opinion, is most suited to school work.

Students may work on assignment or activity sheet to review Bank Street Writer.

ASSIGNMENT 12/13

Word Processing

INSTRUCTIONS Using the Bank Street Writer, enter the following two paragraghs exactly the way they appear here - including errors!

Even though it looks almost the same as the Apple 2, the Apple 2e gives you a great deal more for your money. The base-priced machine includes 64K bytes of memory, Applesoft BASIC, and a 63-key keyboard, seven expansion slots, and a video interface that can display 24 lines in a 40 column format.

For about the same price as the Apple 2, the Apple 2e provides a variety of exciting new features and capabilities. Rather than start from scratch and design a new machine, Apple Computer Inc. chose to make a series of enhancements and improvements while keeping the style of the Apple 2. Although redesigned, the 2e is clearly a member of the Apple family.

INSTRUCTIONS After you enter the previous two paragraphs exactly as they appear, the object of this lesson is to test you to see if you have learned how to use the features of the Bank Street Writer. You should now go back to make the following changes:

1. Completely move the second paragraph to the beginning of the text so that both paragraphs are interchanged.

2. Replace all references to Apple 2 or Apple 2e to read Apple II or Apple IIe.

3. Proofread and correct any spelling errors.

4. Print a final copy of the newly formatted text and hand it in for a grade.

ASSIGNMENT 12/13

Word Processing

ANSWER KEY

REFORMATTED TEXT

For about the same price as the Apple II, the Apple IIe provides a variety of exciting new features and capabilities. Rather than start from scratch and design a new machine, Apple Computer Inc. chose to make a series of enhancements and improvements while keeping the style of the Apple II. Although redesigned, the IIe is clearly a member of the Apple family.

Even though it looks almost the same as the Apple II, the Apple IIe gives you a great deal more for your money. The base-priced machine includes 64K bytes of memmory, Applesoft BASIC, a 63-key keyboard, seven expansion slots, and a video interface that can display 24 lines in a 40 column format.

TO THE INSTRUCTOR

Look for these important items in grading this document:

1. Were the paragraphs completely moved?

2. Are both paragraphs still properly indented?

3. Were all 2's replaced by II's? On this point you have to check to see if the 2 in the "24" of the last sentence was improperly changed. The program assumes that it is a candidate to change and if the students is not paying close attention they will make this mistake.

BANK STREET WRITER TEST

Type the following two paragraphs, including any errors:

> This is one example of a use for a computor within the skool or home. This is a user freindly word processing program. It is called Bank Stret Writr. Eagles are birds.
>
> There are many uses for computors in our everday lives. It certainly doesnt take a programming wizard or mathematical genius to use a computer.

YUOR TEST.....

1. Switch the two paragraphs around.
2. Change the spelling of COMPUTOR to COMPUTER and correct any other spelling errors.
3. Delete any word or sentence that doesn't make sense.
4. Save the text on an initialized diskette as your name (the first eight characters).
5. Retrieve the text, if necessary.
6. Print the draft copy.
7. Submit draft copy to your instructor.

Computer Education

BANK STREET WRITER TEST I

YOUR TEST.....

You will have 10 minutes to:

1. Load the text that has been filed as BSWTEST.
2. Switch the two paragraphs around.
3. Change the spelling of COMPUTOR to COMPUTER.
4. Correct any other spelling errors.
5. Remove any word or sentence that doesn't make sense.
6. Transfer the edited text to the diskette as YOUR LAST NAME (the first eight characters).
7. Remove your text from the screen.
8. Notify your instructor that you are done.

GOOD LUCK.....GOOD WRITING.....

COMPUTER EDUCATION

ASSIGNMENT SHEET (Bank Street Writer)

COPY THE FOLLOWING QUESTIONS INTO YOUR NOTEBOOK. DO NOT WRITE ON THIS SHEET.
BE BRIEF WITH YOUR ANSWERS, BUT BE THOROUGH.

1. What does RETRIEVE mean? What does it do?
2. What is the cursor?
3. What is Bank Street Writer an example of?
4. What does MOVE mean? What does it do?
5. Explain what you have to do to move the cursor around the text without erasing?
6. When entering text, what does CONTROL-C allow you to do? Explain.
7. Explain how you save your text onto the disk.
8. What is the CATALOG?
9. Can you guess why a password is provided for when saving or retrieving text?
10. What is a MENU? Explain.
11. What does ESC refer to?
12. What does REPLACE allow you to do? Explain,
13. What does FIND allow you to do? Explain.
14. What is meant by HIGHLIGHT? Explain how it is used within Bank Street Writer.
15. Explain the proper handling of diskettes (the do's and don'ts).
16. Explain the two ways in which you can change "screens" with Bank Street Writer.
17. With the CAPS LOCK Key unlocked, explain how you type upper case letters or upper key characters.
18. When writing on the text screen, what purpose do the left and right arrow keys serve?
19. What is meant by CLEAR? Explain.
20. What is meant by QUIT? Explain.

COMPUTER EDUCATION

Activity Sheet (Bank Street Writer)

ANSWER THE FOLLOWING QUESTIONS. GIVE YOUR ANSWER IN OUTLINE FORM. WRITE BRIEFLY, BUT CLEARLY. DO NOT LEAVE OUT ANY DETAILS.

1. When the program for Bank Street Writer is loaded into the computer, you are at the WRITE screen. A disk is given to you which has your text stored on it. List all of the steps necessary for you to load the contents of the disk, make corrections and/or additions, save the newly edited text onto the disk, and get the computer ready for the next student. Don't leave out any details.

2. You are at the TRANSFER MENU screen. You have just saved a text file and wish to start another text file. List all the steps necessary to accomplish this.

LESSON NUMBER 14

TITLE Word Processing Project

GOALS

Problem-solving Tools : Students will value and use computers as general-purpose machines designed, built and operated by humans as problem-solving tools.

Basic Computer Operation : Students will become self-sufficient in the basic operations of the keyboard, peripheral devices and software manipulation.

LEARNING OUTCOMES

1. Students will efficiently use a word processing program to complete an assignment or term paper.

2. Students will edit and format their paper as well as print two copies of a final draft. One copy will be handed in to the instructor for whom it was due and one copy for this lesson.

BACKGROUND INFORMATION

Independent use of the computer to load, run and properly handle software and equipment can not really be evaluated without having students function completely on their own to solve a problem.

REFERENCE MATERIALS

See lesson 12/13 and the "Bank Street Writer" support booklet.

MATERIALS NEEDED

1. "Bank Street Writer" program disk

2. "Apple Writer" program disk

3. Blank disks as necessary

LESSON DESCRIPTION

1. Students are clearly on their own and should be given great latitude in choosing an appropriate topic for use in this lesson.

2. Once an assignment is chosen, students should be given the entire class period, and possibly more, to complete the project.

3. The assignment should be printed and copies handed in to both instructors for grading.

A BEGINNER'S GUIDE TO

APPLE WRITER

(an excerpt from: Modular Computer Lesson Design 2nd. ed.
by Paul M. Roper and David V. Lowetshcher)

Hi Willow Research and Publishing

c. 1984

WORD PROCESSING WITH APPLE IIe WRITER
-- a short guide by David V. Loertscher --

Equipment - an Apple IIe computer with an 80 column text card inside the computer, at least one disk drive, a monitor, one disk marked Apple Writer II, and one blank disk.

The purpose of this short guide is to introduce the most commonly used features of Apple Writer IIe. Using this guide will prepare you to consult the Apple Writer Manuals with more understanding. Our objective is to help you get your feet wet.

Getting Started

1. Put the Apple Writer disk in Drive One (label side up) and close the door.

2. Boot the disk by turning on computer (adjust monitor if necessary).

3. After Copyright Screen comes on, press RETURN and wait until red light on drive goes off. The word processing program is now available in your computer.

4. Remove Apple Writer disk from drive and insert blank disk to initialize it.

5. Type CTRL O (Hold down the key marked CONTROL and type the letter O).

6. Type G then type S6,D1 and press RETURN and wait until red light goes off.

7. Press RETURN to exit - you are read to "process words".

8. You now have initialized a storage disk for your files. The next time you start with Apple Writer you will not do steps 4-6. Instead, you will insert your storage disk after step 3.

Writing on the Screen

Your blank screen is now an electronic piece of paper. The blinking cursor is your electronic pencil. The strip across the top of the screen is the data line. It contains seven pieces of information that will be helpful later on.

Use the keyboard much as you would a typewriter including the shift key for capital letters and caps lock to make all caps. The biggest difference is not having to use a carriage return at the end of each line. As you type a line, the program will push words on to the next line when it gets too full -- it's magic -- it's called word wrap around. Each line will take up to 80 characters (including spaces).

Do the following:

1. Push the control key and the letter Q down at the same time. This should give you the "Additional Functions Menu". Now push G. Your cursor now has

a bracket inside it. When you push the return key from now on, it will leave a bracket there. This will show you where your carriage returns are. After you get used to word wrap around you can turn off the bracket by pressing CONTROL Q and G again.

2. Hold the space bar down and notice how the position counter (POS) works. On what number does the cursor return to the left line? (answer: 80)

3. To get rid of all those spaces you just made on the screen, press DELETE and hold it down until the POS number returns to zero.

4. Now type in the following joke but do not type in a carriage return until you are at the end of the joke. Put five spaces on your screen to indent the first line of your joke.

What did the baby porcupine say when it backed into the cactus? Is that you, Mother?]

Does your joke look just like the one above? If you made any mistakes or you have a carriage return you should not have, use the DELETE key to erase and do it over again.

Now add a new joke as a new paragraph:

What is black and shiny and lives in trees and is dangerous? A crow with a submachine gun.]

5. You can move your electronic pencil (the cursor) around by using the arrow keys. Try skipping through the text with the four arrows. You can move one letter at a time by pressing the arrow once. You can move faster by holding the arrow key down. You can skip even faster by holding down the FILLED APPLE and the arrow keys. Get your cursor back to the end of the second joke.

6. We can add words in the middle of sentences as if by magic. Let's do it. Use the arrows to move the cursor until it is on the first "b" of "baby" in the first joke. Now type the word "little" and add a space at the end. Notice how all the words in the paragraph are pushed aside to make room for the new word. Now add these words to the second joke: <u>usually</u> lives ; in <u>big tall</u> trees.

7. Now you need practice in deleting and adding. The DELETE key only works from right to left. In the first joke, delete the word "little" by putting your cursor on the first letter of "baby" and deleting left. Now delete the words "big tall" in the second joke.

8. Now add a new joke right as a new paragraph between the other two jokes:

Why does an elephant have a trunk? So that it has someplace to hide when it sees a mouse.]

9. Now get your cursor at the end of your document - CONTROL E is the fastest way. Look for a little arrow at the left of the data line at the top of your screen. Change the direction of this arrow by pressing CONTROL D. It should be pointing left. Now press CONTROL X. You just erased your third joke. Press CONTROL X twice more and your screen will be blank. You now know how to erase full paragraphs.

10. Now you are ready to do fancy things. Type a friendly letter to someone you know. Use a date in the upper right hand corner and indent as you should. Make your letter about 50 words long.

11. Add a sentence in the middle of the letter.

12. Proofread your letter and correct your errors.

Summary

CAPS LOCK -- when down - all uppercase letters
-- when up - all lowercase letters

SHIFT -- use to get uppercase letters when CAPS LOCK is up

4 arrow keys (up, down, left, right) -- move the cursor without erasing anything

DELETE -- deletes letters from <u>right</u> to <u>left</u>

Filled Apple and arrow keys -- move cursor faster

RETURN -- returns cursor to beginning of next line - not needed until the end of a paragraph of section

CTRL B -- go to the beginning of the file

CTRL E -- go to the end of the file

Saving and Loading Files to and from the Disk

1. Type CTRL S and then a name for the file (such as Letter) and press RETURN.

2. To make sure the file is saved, type CTRL O and select A for CATALOG and press RETURN. Is your file on the disk? If not, get help. To get back to your file, press RETURN, and press RETURN again. Notice that the file name

has been added to the data line. If you are writing a long document, you will want to save every 15-20 minutes (this is insurance against loss).

3. Before loading another file, type CTRL N and then Y to get the old file off the computer's screen. If you don't clear your screen before loading a new document, you will have two documents on your screen - the old and the new.

4. To load a file type CTRL L and the name of the file (such as Letter) and press RETURN. The Mem on the data line shows how much space is left on that file. The Len shows how many characters are presently stored in that file.

5. Resaving is done after editing or making additions. It can be done under the old name or a new name. A shortcut to resave the old name file is to use CTRL S =. This will save the file with the name exactly as it is shown on the data line. If you use a new name then you will have two files, one with the old name and one with the new name.

6. To delete a file, go to the DOS commands by using CTRL O.

Summary

CTRL S -- save a file onto a disk

CTRL S = -- save a file under the name up in the data line

CTRL O -- DOS commands (catalog, rename, verify, lock, unlock, delete, or initialize)

CTRL N -- clear file off the screen

Printing Out a File

You are now ready to print your letter out on the printer.

1. Turn on the printer.

2. Now you need your letter on the screen. If it is there, skip step #3.

3. Press CONTROL L. It will ask for a file name. Type in the name of your file (I hope you remember what you named it - if not type CONTROL O and option A to see what names are on the disk). Press RETURN. Your letter should be on the screen.

4. Press CONTROL P

5. Type np -- The document will be printed out on the printer. np means new printing.

Summary

CTRL P -- gets into the print mode.

? -- gives you a list of formatting codes (how you want the document printed).

np -- prints out the document on your screen starting at the beginning

cp -- continues to print on to the end of a previous document.

Getting Better at Word Processing

You now know the rudiments of word processing. You might want to practice creating and printing simple documents several times before you go on. The rest of this document shows some common things you can do with Apple Writer. Look through the headings so that you know what you can do. When you need to add paging or move paragraphs, go to that section of the handout and practice. Soon you will need to know more advanced techniques to make your word processing easier. You should now be able to read the Apple Writer manuals with some understanding.

Numbering Pages

Apple Writer will start counting on page 1 with number 1 unless you give it a different number for the first page. To do that, change the PN command on the print command list to the number you want.

For placement of numbers, begin with CTRL P and ? followed by:

TL///#/ -- top right corner of the top of the page

TL//#// -- top middle of the page

TL/#/// -- top left corner

BL///#/ -- bottom right corner

BL//#// -- bottom middle of the page

BL/#/// -- bottom left corner

After typing one of the above, press RETURN. Did the display change? If not, try again.

Replacing Words or Phrases

Checking again for errors.

1. Use CTRL B to get to the beginning of the file.

2. Use CTRL D to point the beginning arrow on the data line in the direction you are checking.

3. Use CTRL F to find words you want to check for spelling, or usage, etc. Put a slash mark at the beginning and end of the word or phrases you want to check. Example -- /Sincerely/

4. CTRL F /mispelled word/correctly spelled word/ -- will find the mispelled word or phrase and ask if you wish to replace it with the correctly spelled word. Y - replaces. RETURN - looks for the next occurrence. Any other key - cancels.

5. CTRL F /mispelled word/correctly spelled word/a -- will replace all occurrences of the misspelled word or phrase in the document.

6. CTRL F <>< and RETURN will make the cursor jump from paragraph to paragraph.

7. CTRL R will let you type right over the top of any character - won't insert any spaces. This is really helpful when creating columns.

Moving and Deleting Text

When you use the DELETE key, everything you delete is gone forever. Fortunately there are ways of temporarily deleting material which can then be retrieved when needed.

1. Use CTRL D to change direction of the arrow on the data line. Left is for temporarily deleting, right is for retrieving.

2. For moving a few words (less than a paragraph), hold down the Open Apple key and press the left arrow key. 128 characters can be temporarily hidden this way. To retrieve the hidden words, position the cursor where you want the words placed, change data line arrow to the right, and hold down the Open Apple key and press the right arrow key until they all reappear.

3. CTRL X deletes and retrieves a paragraph from right to left. Change data line arrow to the left, position cursor to the right of the parts to be temporarily deleted, use CTRL X, move cursor to place for insertion, change data line arrow to the right, and press CTRL X.

4. To copy a paragraph and duplicate it somewhere else, position cursor at end of part to be copied, change data line arrow to the left, hold down the Filled Apple key and use CTRL X. Then move cursor to desired place for insertion, change data line arrow to the right and use CTRL X for retrieval.

Word Wraparound

CTRL Z -- Turns on and off the word wraparound function (automatic carriage return). The Z will appear at the top of the screen if wraparound is ON.

Underlining

To underline text type the backslash at the beginning and end of what you want underlined. The backslash is printed out as a space at the end of the word, so do not put a space after what is being underlined.

Example:

The name of the book is \Morning is Nigh\and it's funny.

produces:

The name of the book is Morning is Nigh and it's funny.

Centering a Line

Use .cj in the text on a line all by itself at the left margin and just before the line you want centered. Then type .lj on the next line to stop the centering. Example:

```
.cj
This line will be centered
So will this line.
.lj
This line will not be centered.
```

Setting and Clearing Tabs

Apple Writer automatically sets tabs for you at every eight spaces. Use the TAB key to move the cursor nine times across the screen. Notice the number that appears on the data line after tab.

You can also set and clear your own tabs.

1. CTRL T and typing P will purge the existing tabs.
2. Use the space bar to position the cursor where you want TABS.
3. CTRL T and typing S will create a tab position, repeating for each setting.
4. CTRL T and typing C will clear the tab where your cursor is.

Making and Using a Glossary

This is a joy. A glossary allows you to type common phrases and sentences with one keystroke. You store the phrases and then retrieve them anytime.

Building a glossary:

1. CTRL N (clear memory).

2. Type in definitions (composed of a one letter designator followed by the definition).

Example:

```
aacounts payable
sSincerely yours,
1We love you very much.
```

Any key on the keyboard can be the designater. Even upper case letters can be designators.

Saving a glossary:

1. CTRL S and name the glossary (glossary is a good name).

Using a glossary:

1. CTRL L to load your document.

2. CTRL Q, choose E, and type the name of your glossary. When you press RETURN, nothing happens on the screen but your glossary is stored and is ready to use.

3. As you are typing along and you need a phrase from the glossary, just press CTRL G and type the letter which represents the phrase.

Example: CTRL G a prints accounts payable.

Changing the Way a File is Printed

1. CTRL P puts the computer into print mode.

2. Type ? to get a list of Print/Program Commands. The values listed for each command are the "default" values -- the ones that Apple Writer will use to print text unless you change the values.

3. To change any one of the command values, type the two letter code and the value you want changed. No space or = sign is needed.

Set left margin	LM ___	default 9
Indent paragraph margin	PM ___	default 0
Right margin from left edge	RM ___	default 79
Top line margin printed under page #	TM ___	default 1
Bottom margin before footer	BM ___	default 1
Set page number	PN ___	default 1
# of printed lines per page	PL ___	default 58
# of lines from top of one page to next	PI ___	default 66 (assumes 11" paper)
# of spaces between lines	LI ___	default 0 (single space) 1 = double space 2 = triple space
Single or continuous paper	SP ___	0 = continuous paper 1 = single sheets fed in one at a time
Printed on screen or printer	PD ___	0 = screen 1 = printer
Carriage return on	CR ___	default 0 1 = add a carriage return

4. Turn printer switch on and type np (for new print) to cause the text of the document in memory to be printed.

5. To print a document at the position on the page where the last one ended, type cp (for continuous print).

To change print commands within the document (for example, if you want some lines single spaced within a double-spaced document) then the commands listed above must be embedded or contained within the body of the text itself.

1. Carriage return to begin a new line.

2. Type a period on the left margin and the command wanted.

3. Carriage return.

Examples:
.li0 -- would turn on single spacing
.li1 -- would turn double spacing on again
.ff -- Form Feed. This will eject the paper from the printer and start printing on a new sheet.
.in -- Input. This will stop the printer and will display a message on the screen, waiting for you to press return before it continues printing.

Printing Out Only Part of a Document

Use the embedded command .ep (enable print) to print out specific parts.

.ep0 -- placed at the beginning of the part you don't want printed (on the left margin always).

.ep1 -- placed at the beginning of the part you do want printed. This means you must put an .ep0 at the end of what you want printed so that printing won't continue beyond that part.

Another way to print out only part of a file is to load only a part of the file first. This is done by specifying the beginning words and ending words of the part you want. You must specify enough words to distinguish that section from any other section.

1. Type CTRL L and file name followed by a comma and d1 (for drive one) then a slash mark used as a delimiter followed by the specific beginning words, and another slash mark followed by the specific ending words followed by another slash mark.
   Example:
   CTRL L Weather,d1/Today is gloomy/much effort./

   These commands cause the paragraph chosen to be loaded, then the CTRL P command can be used along with np to get a printout of that paragraph.

Printing Several Documents as One

1. CTRL N to clear memory.
2. CTRL L and name first document.
3. CTRL P and np to print.
4. CTRL N to clear memory.
5. CTRL L and name second document.
6. CTRL P and cp to print next document.

Printing to the Screen - Not to the Printer

You can check your document before you waste printer paper.

1. CTRL L and name file.
2. CTRL P and ?
3. Change command PD (print destination) from 1 to 0.
4. Print out using np.
5. Use CTRL S to stop and continue scrolling.

LESSON NUMBER 15

TITLE Trade Applications

GOALS Impact on Jobs : Students will realize the impact of computerization on the job market as well as specific trade areas.

General Uses : Students will become aware of the variety of computer applications, their advantages, disadvantages, limitations and impact on individuals and organizations.

LEARNING OUTCOMES

1. Students will develop an understanding of the significance computers will play in many if not all of the shop areas.

2. Students will be able to name several computer applications in the shop areas.

BACKGROUND INFORMATION In order to be fully prepared for a career, students must be fully cognizant of all equipment, tools and skills that they may encounter on the job. At the very least, students should be aware enough of these possibilities to be flexible when they are confronted with innovative equipment.

REFERENCE MATERIALS Computer Literacy: Problem Solving with Computers , pages 45 to 66

MATERIALS NEEDED Samples of any equipment listed in the lesson description that are available in the school

VOCABULARY microprocessor, CAD, CAM, robotics

LESSON DESCRIPTION

I. Contrast computer controlled machines with those operated by hand.

II. Discuss and demonstrate wherever possible each of the following shop applications

A. AUTO MECHANICS

1. Microprocessors in monitoring engine performance

2. Computerized engine analysis

3. Computerized inventory of parts

B. AUTO BODY

1. Computer simulation of stress in crash

2. Computerized estimates of parts and labor

3. Computerized inventory control

C. Carpentry

1. Calculating materials and/or labor costs

2. Computerized inventory

3. Accounting

D. DRAFTING

1. Digital drafting machines

2. Computerized piece plotting

3. Telecommunications

4. Computer Assisted Drafting (CAD)

E. ELECTRICAL

1. Monitoring electrical flow

2. Power requirements for mainframes

3. Parts inventory

F. FOOD SERVICE

1. Food inventory

2. Menu storage

3. Accounting

4. Computer controlled microwaves

G. INDUSTRIAL ELECTRONICS

1. Logic circuits

2. Microprocessors

3. Binary - machine language - octal - hex.

G. HAIRDRESSING & COSMETOLOGY

1. Computerized cash registers
2. Accounting systems
3. Computerized hair color analysis

H. MACHINE

1. N.C. Machines
2. Robotics
3. Digital Control

I. MAJOR APPLIANCE REPAIR

1. Microprocessors as integral parts of major appliances
2. Programmable major appliances
3. Parts inventory

J. PLUMBING

1. Parts inventory
2. Cost estimating
3. Accounting

K. PRINTING

1. Word processing
2. Disk storage
3. Digital control machines
4. Typesetting
5. Computer graphics

INDEPENDENT ACTIVITIES

Students should select one computer application in their preferred shop area and prepare a short class report explaining the uses and skills needed to operate the equipment.

Students may run Mico Cookbook, Hi-Res Architectural Design, MECC 918 Industrial Arts (Micrometer, Rafter).

LESSON NUMBER 16

TITLE Uses and Abuses of Computers

GOALS

Ethics : Students will understand the importance of behaving in an ethical and responsible manner in relation to computers and information systems in society.

General Uses : Students will become aware of the variety of computer applications, their advantages, disadvantages, limitations and impact on individuals and organizations.

Impact on Jobs : Students will realize the impact of computerization on the job market as well as specific trade areas.

LEARNING OUTCOMES

1. Students will be aware of the role of computers in medicine, government, business, farming, education, manufacturing and design.

2. Students will be aware that half the U.S. economy is concerned with information processing.

3. Students will understand that future changes in information handling will give rise to electronic mail, electronic publishing and the electronic cottage.

4. Students will be aware of the dangers from invasion of privacy, computer theft, computer fraud and job displacement.

BACKGROUND INFORMATION

There are an estimated five billion computers in existence for a world population of four billion. This includes the chips in watches, appliances etc. It is estimated that hundreds of thousands of people are discouraged from voting in national elections in the western states because of early computer projections of winners based on East coast balloting. Finally, there is a periodic attempt to establish a national data center. There is a great deal of current material in newspapers and magazines that can be used for this area.

REFERENCE MATERIALS

1. Are You Computer Literate , pages 91 to 111
2. Current local newspapers.

VOCABULARY

UPC, EFT, electronic mail, electronic publishing, electronic cottage

LESSON DESCRIPTION

1. Survey students for computer applications they are familiar with, e.g., automatic tellers, credit cards UPC, watches, appliances...
2. List categories of use - medicine, government, business, farming, education, manufacturing and design. Discuss applications in each category.
3. Discuss threats to privacy through credit checks, EFT (Electronic Fund Transfer), UPC and census results.
4. Discuss the effects of computer errors on innocent individuals.
5. Discuss computer crimes.
6. Discuss job loss and job control because of computers.
7. Discuss future uses of electronic mail, electronic publishing and the electronic cottage.

INDEPENDENT ACTIVITIES

1. Assign thought question, "Are You a Computer Literate".
2. Have students clip articles about positive and negative impacts of computers from local newspapers.

ASSIGNMENT

From "Are You Computer Literate?"

by Karen Billings and David Moursund

Dilithium Press - 1979

Thought Question pg. 104:

"Make a list of all the places where there are records containing detailed information about you (school records, medical records, etc.). What is in these records? Who has access to these records? Would you like it if all your friends (and enemies) had easy access to all these records? Discuss why or why not.

It is sometimes suggested that we are becoming a "cashless" society. More and more goods and services are paid for by check or credit card. What are the obstacles to completely stopping the use of cash? What would the effects be: Would you like this to happen?

Discuss the situation and the likelihood that it will happen in the next 20 years."

LESSON NUMBER 17

TITLE Computers and Careers

GOALS

Impact on Jobs : Students will realize the impact of computerization on the job market as well as specific trade areas.

General Uses : Students will become aware of the variety of computer applications, their advantages, disadvantages, limitations and impact on individuals and organizations.

LEARNING OUTCOMES

1. Students will be able to identify main categories of computer related careers, ranging from the design and manufacture through application and operation.

2. Students will be able to describe specific jobs in each area, the nature of the work itself, and employment outlook and earning potential.

3. Students will be able to outline the necessary education and job training as well as qualifications needed with each position.

BACKGROUND INFORMATION

For students taking part in an exploratory phase in their schooling, an awareness of the most promising career areas available to them is vital for success and personal growth. For those students whose interests and abilities suggest it, one of the most promising areas for career development is in the field of computers.

The purpose of this lesson will be to expose the student to a wide variety of computer-related job applications in the specific areas of 1) Hardware 2) Software and 3) Operations.

Students and instructor may consider asking the Guidance Department in their school to demonstrate the CONSIDER/GIS program for more information on career areas of interest to them.

REFERENCE MATERIALS — Spotlight on Computer Literacy , pages 69 to 72

MATERIALS NEEDED

1. Slide projector, screen, cassette recorder
2. "Careers in Computer Science and Service" slide presentation parts 1 through 4.

VOCABULARY — hardware, software, operations, computer technician, systems engineer, systems analyst, computer-operations manager, electrical engineer, programmer

LESSON DESCRIPTION

1. Present slide program.
2. Discuss the many careers presented in the program. Which ones might be appropriate for technical school graduates, which ones would require more education?
3. Discuss the interaction between people who work directly in the computer field and workers in the trades the students are interested in.
4. Discuss the computer related skills that the students will need even though they may not work in the computer field.

INDEPENDENT ACTIVITIES — Complete activities on page 29 of the "Careers in Computer Science and Service" teacher's guide.

LESSON NUMBER 18

TITLE Programming Logic

GOALS

Programmming Logic : Students will understand and apply simple logic, principles and techniques related to computer programming.

Problem-solving Tools : Students will value and use computers as general-purpose machines designed, built and operated by humans as problem-solving tools.

LEARNING OUTCOMES

1. Students will identify four computer languages: BASIC, FORTRAN, COBOL, and Pascal.
2. Students will identify the major area of use for each language.
3. Students will be able to list the five logical steps for solving a problem on the computer.
4. Students will be able to develop algorithms for the solution of non-computer problems.

BACKGROUND INFORMATION

It may be necessary to briefly review the binary operation of computers at the start of this lesson. At least, the definitions of bit and byte should be reviewed.

REFERENCE MATERIALS

1. Secret Guide to Computers , Vol. 8
2. Are You Computer Literate
3. Nailing Jelly to a Tree
4. Fundamentals of Digital Computers
5. MECC Beginning Applesoft Basic , pages I-1 to I-6
6. Apple IIe Owner's Manual , pages 87 to 91
7. "Computers: Peripherals to Programming" filmstrip program

MATERIALS NEEDED

1. Overheads of MECC pages I-8 and I-9, attached
2. Copies for each student of MECC activities I-11 and I-12, attached

VOCABULARY program, algorithm, BASIC, FORTRAN, COBOL, Pascal, flowchart, data

LESSON DESCRIPTION

1. Discuss the importance of specific and accurate instructions in the trades, (blueprints, shop manuals on repair and assembly and trouble shooting).
2. Discuss recipes in terms of logic, sequence and final product.
3. Discuss the types of instructions required by computers, (add, divide, multiply, subtract, operate peripherals, store data, compare numbers).
4. Stress the fact that computers respond only to coded instruction in terms of 0/1, on/off, or magnetized/non-magnetized.
5. Discuss the acronyms and uses of the following languages: BASIC, FORTRAN, Pascal and COBOL.
6. List the steps involved in writing a computer program: Define problem, Develop a solution (algorithm), Draw a flowchart, Write a program, Test and debug.
7. Elicit group solutions to MECC overheads I-8 and I-9.

INDEPENDENT ACTIVITIES

1. Assign MECC activity page I-12.
2. Assign MECC activity page I-11, Activity 2.

COMPUTER EDUCATION
ACTIVITY SHEET (logical sequence)

IN EACH OF THE FOLLOWING ACTIVITIES, A SET OF STEPS HAS PEEN PREPARED. ARRANGE THE SET SO IT IS IN LOGICAL ORDER.

1. CALLING A FRIEND--
   a. dial a friend's number
   b. hold a conversation
   c. end call--conversation completed
   d. tell friend goodbye
   e. identify yourself
   f. dial friend's number again
   g. friend answers telephone
   h. end call--line is busy

2. MAKING A SANDWICH--
   a. place two pieces of bread together, peanut buttered sides touching
   b. remove two slices of bread
   c. pick up knife
   d. use knife to get some peanut butter from jar
   e. spread peanut butter on bread using knife
   f. continue spreading peanut butter until both slices of bread are covered
   g. open jar
   h. open bread package
   i. get peanut butter, knife and loaf of bread
   j. eat and enjoy

3. SOCKS--
   a. put on shoes
   b. pick pair of socks
   c. put on socks
   d. check if socks match
   e. walk away
   f. go to dresser

4. KEY--
   a. try the key in the lock
   b. open the door
   c. stop
   d. select a key
   e. does the key turn?
   f. start
   g. try to turn the key
   h. does the key fit?

5. PENCIL--
   a. stop
   b. put it in the sharpener
   c. look at the point
   d. put the pencil on your desk
   e. take a pencil
   f. do you have more pencils?
   g. turn the handle
   h. is it sharp enough?
   i. remove the pencil
   j. start

6. DINNER--
   a. eat all vegetables
   b. stop
   c. wash hands
   d. is the plate clean?
   e. take place at table
   f. start
   g. take some vegetables

7. BREAKFAST--
   a. place egg in frying pan
   b. get egg
   c. turn on stove
   d. check if done
   e. crack open egg
   f. turn off stove
   h. get frying pan
   i. place frying pan on stove
   j. place egg on dish
   k. eat egg
   l. remove egg from pan

8. MAKE SPAGHETTI
   a. open the package
   b. is there enough?
   c. fill it with water
   d. start
   e. pour spaghetti into boiling water
   f. drain off the water
   g. boil a big pot
   i. is it al dente?
   j. stop
   k. time the spaghetti
   l. taste the spaghetti

LESSON NUMBER 19

TITLE Flowcharting

GOALS

Programming Logic : Students will understand and apply simple logic, principles and techniques related to computer programming.

Problem-solving Tools : Students will value and use computers as general-purpose machines designed, built and operated by humans as probles-solving tools.

LEARNING OUTCOMES

1. Students will identify the four major flowchart symbols.
2. Students will predict the outcome of a simple flowcharted program.
3. Students will break a simple program into logical sections and make a flowchart.

BACKGROUND INFORMATION

The instructor should be aware of the meaning of all the symbols on the flowchart template even though only the four principle symbols will be used. It should also be noted that these symbols are called ANSI (American National Standards Institute) symbols.

REFERENCE MATERIALS

1. Are You Computer Literate
2. I Speak Basic to my Apple
3. Basic Basic
4. MECC Beginning Applesoft Basic , pages I-7 to I-13

MATERIALS NEEDED

1. Overheads from MECC "Beginning Applesoft Basic", pages I-7, I-13, I-9 and I-10 attached

2. Copies of activity page I-11, attached

VOCABULARY

flowchart, terminal symbol, process symbol, decision symbol, input-output symbol, loop, branching

LESSON DESCRIPTION

1. Review five basic steps for solution of problems.

2. Discuss student solutions to assignment for lesson 18, page I-11 Activity 2.

3. Define flowchart and give definitions for the four basic flowchart symbols.

4. Discuss with the class the steps required to make a long distance telephone call. List the steps and have the class determine the proper flowchart symbols for each one.

5. Present overhead I-13 and arrive at a group solution.

6. Present overhead I-11. Introduce the concept of "playing computer" and, using this technique, predict the outcome of the program.

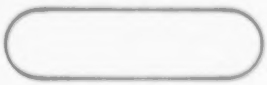
INDEPENDENT ACTIVITIES

Assign MECC activity page I11 activities 3, 4 and 6.

Step three of the program development process involves drawing a flowchart. Besides providing a diagram of a solution to a problem and a guideline for the preparation of a computer program, a flowchart also provides a source for locating potential errors in the program.

Most flowcharts, sometimes flow diagrams, may be prepared using the first four of the following flowchart symbols:

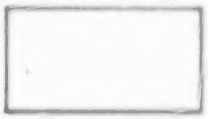
TERMINAL SYMBOL-- marks the beginning and end of a program, or the starting or ending point of a flowchart.



INPUT/OUTPUT SYMBOL-- used when imputting data or printing an output; used to indicate a READ, INPUT and PRINT instruction

DECISION SYMBOL-- used a a decision point in a program; used to indicate a decision

PROCESS SYMBOL-- used whenever other symbols don't apply; indicate processing of data; used to indicate operation or assignment



CONNECTION SYMBOL-- used as a match mark for a flowchart if it is so long that it doesn't fit on page and must be continued on another

ARROW-- used to indicate the direction of flow of the process

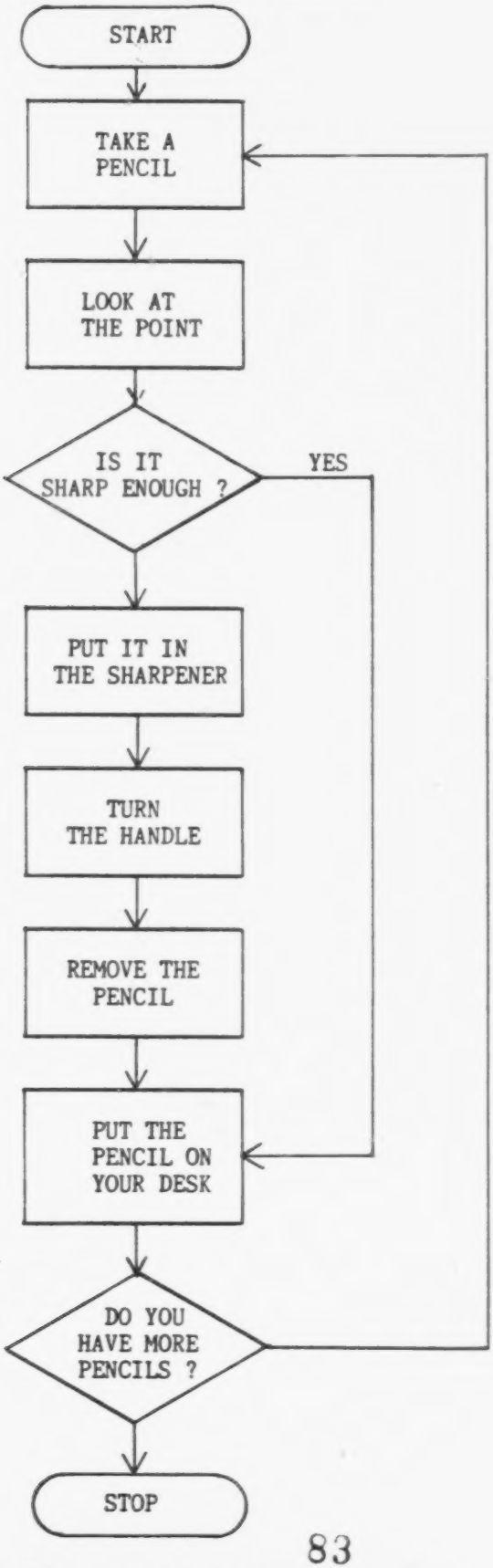
PREPARE A FLOWCHART OF EACH OF THE FOLLOWING, USING THE PROPER SYMBOLS:

1. Start. Read A. If A is less than 3, print the values of 3-A and end. If A is not less than 3, print 3+A and end.

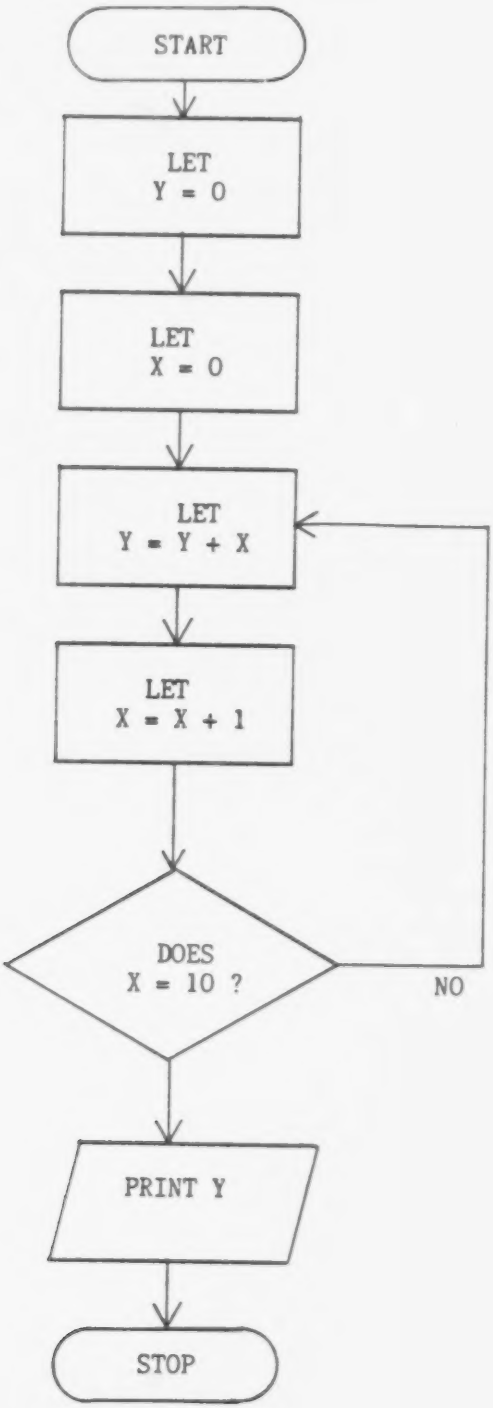
2. Refer to ACTIVITY SHEET (logical sequence) and prepare a flowchart for each of the seven activities, without adding any steps.

3. Repeat activity #2, adding any steps to the flowchart that you feel may be missing.

FLOWCHART FOR SHARPENING PENCILS



FLOWCHART FOR THE SUM
OF THE FIRST TEN INTEGERS



ACTIVITIES

1. Complete the flowcharts on pages 12 and 13.

2. Flowchart the procedure for <u>one</u> of the following activities:

    a. Running a particular play in football
    b. Emptying the garbage
    c. Replacing a burnt out light bulb
    d. Mailing a letter
    e. Using a vending machine

3. Flowchart the procedure for determining the tallest person in a room.

4. Flowchart the procedure for the following problem:
   Jim's mother wants him to -

    1. pick up his little brother who is either at the Brown's or the Williams' house

    2. leave a package at the Olson's if Mrs. Olson is home (otherwise he is to bring the package home)

    3. take care of his little brother until his mother comes home from the beauty shop.

5. Flowchart the six step program development process.

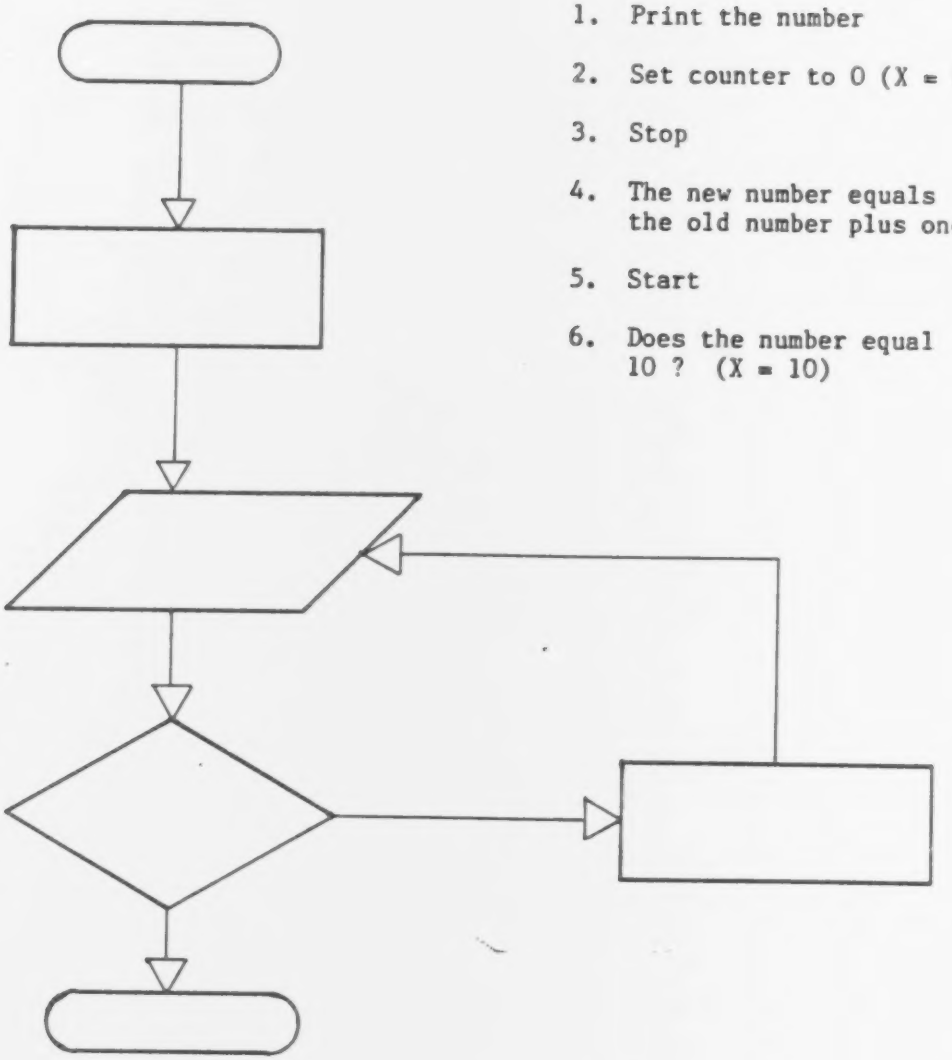
6. Complete steps 1-3 pf the program development process (define problem, develop algorithm and draw flowchart) for <u>one</u> of the following problems:

    a. Given any integer, determine whether it is positive, or negative, or zero.

    b. Assuming that everyone sleeps 1/3 of the time, calculate how many hours a person has slept in a lifetime.

7. Review the procedure for, and initialize a diskette to be used for activities throughout the remainder of the units.

COUNTING TO 10



1. Print the number
2. Set counter to 0 (X = 0)
3. Stop
4. The new number equals the old number plus one
5. Start
6. Does the number equal 10 ? (X = 10)

COMPUTER EDUCATION
REVIEW SHEET

ANSWER THE FOLLOWING. BE BRIEF, BUT BE COMPLETE. YOU MAY REFER TO YOUR NOTES.

1. What is a BYTE?
2. What is SOFTWARE?
3. What is INPUT?
4. What is a PROGRAM?
5. What is a BIT?
6. What is HARDWARE?
7. Name the most widely used INPUT DEVICE and OUTPUT DEVICE?
8. Define ROM.
9. What is a MENU?
10. Define CATALOG.
11. Define RAM.
12. What does 64 K mean?
13. What is "Bank Street Writer" an example of?
14. What does CPU stand for? What does it do?
15. What is the CRT?
16. What is a BINARY DIGIT? a BINARY DEVICE?
17. What does SAVE mean when using "Bank Street Writer"?
18. What is OUTPUT?
19. Name the precautions needed when handling and storing a diskette.
20. What is the MONITOR?
21. What does RETRIEVE mean, when using "Bank Street Writer"?
22. What is the purpose of the "green screen"?
23. When using "Bank Street Writer", when can you freely move the cursor around without erasing? Explain.
24. What is the only language the computer understands?
25. What is one advantage of a disk drive over a cassette? over a disk drive?
26. Whether cassette or disk, how is information stored?
27. What is BASIC?
28. Define ALU. Where is it located?
29. What is an integrated circuit chip? Explain.
30. What is the purpose of a flow chart? Explain.

LESSON NUMBER 20

TITLE Introduction to Hands-on Programming

GOAL Programming Logic : Students will understand and apply simple logic, principles and techniques related to computer programming.

LEARNING OUTCOMES

1. The student will understand that the initializing process writes DOS on the disk, establishes a catalog area and marks the sectors where programs will be stored.

2. Students will know how to initialize a blank disk using the System Master.

3. The student will understand the meaning of the following commands: NEW, LIST, RUN, SAVE, INIT, CATALOG.

4. The student will know the BASIC symbols for addition, subtraction, multiplication, division and raising a value to a power.

5. The student will know the mathematical order of operations.

6. The student will be aware of computer detected errors.

BACKGROUND INFORMATION

It should be emphasized that the intent of these programming lessons is not to actually teach BASIC programming as a distinct discipline, but to show the learner how a computer is controlled from the keyboard and to give the student hands-on experience in control of the computer.

REFERENCE MATERIALS

1. MECC Beginning Applesoft Basic , pages T-1 to T-2, pages II-1 to II-9, pages III-5 and III-14

2. The DOS Manual

3. Apple IIe Owner's Manual , pages 57 to 69

MATERIALS NEEDED

1. One blank disk for each student.
2. System Master disks.

VOCABULARY — System Master disk, command, DOS, INITialize, SAVE, PRINT, NEW, LIST, RUN.

LESSON DESCRIPTION

1. Distribute System Masters and blank disks to each student.
2. Emphasize that the initializing process erases whatever was contained on the disk before beginning to initialize.
3. Explain the process of initializing diskettes with the System Master.
4. Have each student initialize and catalog a blank disk.
5. Introduce the PRINT statement. Have the students copy, run, list, and save the sample program. Discuss the effects of quotation marks, NEW, and END.
6. Introduce the mathematical symbols for addition, subtraction, multiplication, division, and raising a number to a power. Explain the order of operations.
7. Using PRINT and numeric expressions, demonstrate the calculation and literal printing modes.
8. Emhpasize that in immediate mode the students are not writing a program that can be repeated by a RUN command.

INDEPENDENT ACTIVITIES — Assign MECC activity page III-2.

INITIALIZING DISKS

To prepare a new disk for storing programs, you must first initialize the disk. The initializing process writes DOS (Disk Operating System) onto the disk, sets up the catalog area and marks the 560 sectors where programs and files will be stored. Initializing a disk containing programs will wipe it clean of all old programs.

Step 1: With the Apple power turned off, place the DOS 3.3 System Master in the disk drive.

Step 2: Turn on the Apple and wait for the title to appear. It should identify the computer system you are using and the version of Disk Operating System.

Step 3. Take out the System Master and put in the disk to be initialized.

Step 4. Write your Hello program. The following is an example. Press RETURN after each line.

```
NEW
10 HOME
20 PRINT "THIS DISK BELONGS TO ___(your name)_____ "
30 END
INIT HELLO
```

Step 5. Wait the 30 seconds or so until the initializing is finished. The disk drive light will then go out.

Type RUN to see if the program does what it is supposed to do. It should clear the screen (HOME). It should say that this disk belongs to you.

Type CATALOG to see your program listing. You named it HELLO and it is 2 sectors long. The A on the left says you wrote the program in Applesoft BASIC language.

Step 6. Test to see if the new disk will boot or start up the disk operating system. Turn the computer off. Count to 10 slowly. Turn the computer on. Your HELLO program should run and display your identifying message. Your disk is now ready to receive your programs.

SUMMARY OF BASIC STATEMENTS

HOME	Clears the screen and puts the cursor at the top
PRINT	Outputs characters to the screen All characters to be printed must be enclosed in quotes ? can be used instead of the word PRINT
END	Terminates a program. END is optional
REM	Can be followed by any text. Allows programmer to put comments in the program. Nothing following REM on a line is executed
GOTO	Directs control to a specified line number Is often used to establish a loop
, (comma)	Divides output on the screen into three zones Output will be formatted into three columns
; (semi-colon)	Outputs the next characters on the same line, with no spaces between
LET	Assigns a value to a variable. LET is optional
READ	Reads data from a DATA line. Assigns the data to specified variables
DATA	Specifies data, read from left to right, by a READ statement
INPUT	Interrupts execution of the program to request data from the keyboard - to be typed in by program user. Prints a question mark on the screen. Should be accompanied by a prompting message which tells the user what to do
FOR...NEXT	Establishes a loop, a series of program lines, which are executed a certain number of times as defined by the variable limits
IF...THEN	Establishes a condition If the condition is met, the entire statement is executed If the condition is not met, the entire statement is ignored
: (colon)	Can be followed by an additional statement on the same program line Used alone on a line, spaces a program listing
TAB	Spaces to the specified print position on the printer May be used only in PRINT statements

ENTERING, EXECUTING and SAVING a program

To enter a program

1. Insert initialized diskette in the disk drive and turn on the computer.

2. Type NEW. Press RETURN.

3. Type in the program. Press RETURN after each line.

```
NEW
10 REM  SAMPLE PROGRAM
20 PRINT 10/2,  5*5,  20+30
30 PRINT "COMPUTERS ARE FUN."
40 PRINT 400/2
50 PRINT "400/2"
60 PRINT "THAT'S ALL, FOLKS."
70 END
```

To execute a program

4. Type RUN. Press RETURN.

5. LIST the program by typing LIST then pressing RETURN.

6. Notice line 40 outputs '200', whereas line 50 prints '400/2'. Quotes are used around any message, including numbers, that is to be printed, not evaluated. Numbers or expressions, not in quotes, are always processed and the final value is output. The message in line 60 is printed because it was enclosed in quotes.

7. Explain:
   The NEW command clears memory. Memory should be cleared before entering any program, so old program lines don't mix with your new program.

   The REM (remark) statement is ignored by the computer, but tells anyone reading the program what the program is about.

   The END statement is optional but completes the program and informs the computer that this is the end of the program.

To save a program to a disk

Type SAVE followed by a program name.

For example, if we called the above program SAMPLE PROGRAM then we can save it by typing SAVE SAMPLE PROGRAM and pressing RETURN.

Type CATALOG to see if it was saved on the disk.

You should save your work periodically or before you leave the computer. This habit will save much misery if the power goes off or if any number of things occur that can happen to a program in progress.

USING THE COMPUTER AS A CALCULATOR

Sometimes you are not interested in saving your work. You may just want the answers to some quick calculations. Your Apple is a great calculator. And it is quick to work in immediate mode . In the immediate mode you need no line numbers and you do not have to type RUN to get your answers. Just tell the computer to PRINT your answer. When you press RETURN it is there.

If you turn on the Apple with no disk in the drive you can use it in immediate or programming modes because Applesoft BASIC is resident. You cannot, however, SAVE your work.

To use your computer as a calculator just:

1. Flip the switch on.
2. Press CONTROL and RESET, together, to turn the disk drive off.
3. Tell the computer to PRINT your answers.

```
PRINT 4 * 5
PRINT (88+76+57)/3
PRINT 3^2
```

The answers will appear as soon as you press RETURN.

MATHEMATICAL OPERATORS

+ for addition	- for subtraction	* for multiplication
/ for division	^ for to the power of	( ) enclosures

ORDER OF OPERATIONS

There is an established order in which the operators in a mathematical expression are evaluated. Apple conforms to this standard order of operations.

Parentheses are evaluated first, from the innermost out.
Next is raising to a power .
Multiplication and division operations follow.
Last to be processed is addition and subtraction .
Operations of equal priority are performed from left to right.

Thus: 6 + 20 / 4 = 6 + 5 = 11 because division is evaluated before addition.

Likewise; (10 + 20) / 4 = 30 / 4 = 7.5 because parentheses are processed before division.

"Pretty Please My Dear Aunt Sally" is a memory aid for the order: Parentheses, Power, Multiplication, Division, Additions, Subtraction.

Do these examples first on paper then check your answer on the computer.

```
PRINT  2+3*(4-1)-8/4          PRINT  100/5^2+4          PRINT  16-6/2+4*3
```

Immediate mode will print literal messages, also, but you must put them in quotes.
PRINT "THE COMPUTER CAN DO ARITHMETIC." will print that message on the screen.

LESSON NUMBER 21

TITLE Programming in BASIC - 2: PRINT, END, GOTO
Editing a program line, Disk Operating System

GOAL Programming Logic : Students will understand and apply simple logic, principles and techniques related to computer programming.

LEARNING OUTCOMES

1. The student will understand that the main parts of a BASIC statement are the line number, the keyword, and the body.

2. The student will understand the meaning of the following keywords: PRINT, END, GOTO.

3. The student will learn how to correct, add, or delete a program line.

4. The student will use the GOTO statement to control program execution and CONTROL-C to stop an endless loop.

5. The student will understand the concept of the loop.

6. The student will understand the meaning of the following commands: LOAD, DELETE, LOCK, UNLOCK, RENAME.

BACKGROUND INFORMATION In this lesson the students should acquire enough BASIC vocabulary to write a very simple program to display some information on the screen. They should begin to relate hands-on programming with prepared programs they have used.

REFERENCE MATERIALS

1. MECC Beginning Applesoft Basic , pages II-1 to II-3, pages II-6 to III-4, pages III-15 to III-17, pages IV-10 to IV-11

2. A Guide to Programming in Applesoft , pages 1.1 to 1.3, pages 1.10 to 1.11

MATERIALS NEEDED

Copies of the four following pages on PRINT END EDIT, GOTO, DOS commands, Great Program, to each student.

VOCABULARY

program, program line, line number, statement, keyword, argument, body, loop, LOAD, DELETE, LOCK, UNLOCK, RENAME, GOTO, CONTROL-C

LESSON DESCRIPTION

1. Define the three main parts of the BASIC statement: Line number, Keyword, and Argument or Body.

2. Have the students enter an example program from the board or handout.

3. Practice editing the program. Add, change, delete program lines.

4. Explain that an endless loop is a series of program lines repeated over and over.

5. Have the students enter and run a short program with an endless loop. Break with CONTROL-C.

6. Demonstrate the spacing effects of semi-colons and commas.

7. Run through Great Program as a group to exercise the use of DOS commands. Students should predict the results before typing the commands. The words they type are in CAPITAL LETTERS.

INDEPENDENT ACTIVITIES

Assign all or part of MECC activity page II-7.

PRINT END EDIT

A program is a series of step by step instructions that tells the computer to do a task. Each instruction, or program line , consists of three parts:

Line number - usually numbered 10, 20, 30, etc. so there is room to add more lines later. For example you can add lines 12, 15, etc. between lines 10 and 20.

Statement or keyword - tells the computer what to do.

Argument or body - information the computer needs to do the task.

The maximum characters on one program line is 255 or about 6 1/2 lines.

Example:

```
10 PRINT 10 * 2
20 PRINT
30 PRINT "THIS IS A SMALL PROGRAM."
40 END

RUN
20

THIS IS A SMALL PROGRAM.
```

Note:

```
Argument is 10 * 2.
Prints a blank line.
Line # is 30.
END takes no argument.

Command - takes no line #.
#s not in quotes are evaluated.
Result of empty PRINT.
Quotes are not printed.
```

The statement PRINT tells the computer to print something on the screen. Everything, including the spaces you want to be printed, must be inside the quotes. An empty PRINT statement prints a blank line. END tells the computer to stop executing the program. END is optional.

Try entering the program. Make additions, changes, deletions. As you do so you may want to note and practice the following editing techniques:

Correcting an error when entering a program line :

Press RETURN and type the line again correctly.
The new line will replace the old one.

Or use the left arrow. Move back to just before the error.
Type over the error with the correction and move the cursor with the right arrow to the end of the line.

Adding a program line :

Type a new line number and enter your program line.
The computer will put it into correct numerical order in the program.
Type LIST to see the revised listing.

Deleting or erasing a program line :

Type the line number and press RETURN.
Type LIST to see the revised listing.

Usually the computer executes program lines in numerical order, from the lowest to highest. GOTO is a statement which allows the computer to branch back and repeat earlier lines in the program or skip ahead and execute lines further on. GOTO transfers control to the specified line number.

```
NEW                                  Clear memory.
10 PRINT "HAPPY BIRTHDAY!"           Display the message on the screen.
20 GOTO 10                           Go back and execute from line 10 on.
30 END                               End the program. (The program never gets
                                         to this statement.)
```

The above program creates an 'endless loop' - a series of program lines repeated over and over until stopped by the computer user. To stop an endless loop, press CONTROL-C . Hold down the CONTROL key while you press C. The program will 'break' at line 10 or 20. Type RUN to make it go again.

GOTO is often used to do a task over and over. The following program counts by 2. Each time the program loops through line 30, the value of A is incremented by 2. Line 20 prints the new value of A each time. How would you change the program to count by 5, by 10, by 100?

```
NEW
10 A = 0                    Assign 0 to A.
20 PRINT A                  Print the value of A.
30 A = A + 2                Add 2 to the value of A. This sum is the new A.
40 GOTO 20                  Go back to 20. Print the (new) value of A.
```

Punctuation alters output format. For example:

```
Rewrite line 20 with a comma at the end.                What did the comma do?
Rewrite line 20 with a semi-colon at the end.       What did the semi-colon do?
Try:    20 PRINT A;" ";          Did the blank space print between the numbers?
```

Here is one for fun:

```
10 PRINT "(your name)";
20 PRINT " WANTS A DATE WITH ";
30 PRINT "(his/her name)"
40 GOTO 10
```

Note: Make sure you have included the indicated spaces in line 20. Put a space between the first quote and the W and between the H and the last quote. These spaces must be included to separate the words. Remember, put spaces that you need printed, inside the quotes.

DISK OPERATING SYSTEM (DOS) COMMANDS:

CATALOG	Lists the catalog of programs on the disk.
NEW	Deletes any program from memory. Should be used before entering a program into memory.
RUN	Executes the program currently in memory.
HOME	Clears the screen and puts the cursor at the top.
LIST	Lists the program currently in memory.
SAVE (program name)	Transfers program from memory to disk. Must be used with program name.
RUN (program name)	Both LOADs and RUNs the program requested. Transfers program from disk to RAM, then executes it.
LOAD (program name)	Transfers a program from disk to memory. Must be used with the program name.
DELETE (program name)	Deletes a program from the disk unless the program is locked. Must be used with a program name.
LOCK (program name)	Locks a program on the disk so it won't be accidentally changed or erased.
UNLOCK (program name)	Unlocks the program so it can be deleted or changed.
RENAME (old name,new name)	Changes the name of a program on the disk.

EXERCISE - SYSTEM COMMANDS - GREAT PROGRAM

Enter the following program into your computer:

```
10 PRINT 5*3
20 PRINT 12 + 8
30 PRINT 7/2
40 PRINT "JONES"
50 PRINT "5 - 3"
60 END
```

RUN the program.

LIST the program.

LIST 40

LIST 20 - 40

SAVE GREAT PROGRAM

Type NEW

LIST the program. Why was nothing listed?

LOAD GREAT PROGRAM from your disk.

LIST the program. Why is it now listed when it was not before?

CATALOG your disk.

DELETE GREAT PROGRAM

CATALOG your disk.

SAVE GREAT PROGRAM again.

CATALOG

LOCK GREAT PROGRAM

CATALOG - What is different?

Try to DELETE GREAT PROGRAM. Can you? Why not?

UNLOCK GREAT PROGRAM

CATALOG - Any changes?

RENAME GREAT PROGRAM,DOS EXERCISE

CATALOG - What happened to GREAT PROGRAM?

LESSON NUMBER 22

TITLE Programming in BASIC - 3: Graphics
Low resolution and High resolution

GOAL Programming Logic : Students will understand and apply simple logic, principles and techniques related to computer programming.

LEARNING OUTCOMES

1. Students will have experience using low and high resolution graphics statements to display pictures on the screen.

2. Students will know the meaning of BASIC low resolution keywords: GR, COLOR=, PLOT, HLIN, VLIN.

3. Students will know the meaning of BASIC high resolution keywords: HGR, HGR2, HCOLOR=, HPLOT.

4. Students will understand simple graphics capabilities of the computer.

BACKGROUND INFORMATION

In low resolution graphics, students can create a picture out of colored 'bricks'. With 16 colors, low res. pictures show well on the screen.

In high resolution graphics, students can write or draw with colored lines. High res. pictures can be printed using a Grappler interface card and a dot matrix printer.

REFERENCE MATERIALS

1. A Guide to Programming in Applesoft , chapter 6

2. Applesoft Tutorial for IIe Only , pages 13-18, pages 113-118

3. MECC Special Programming Features , Sessions I & II. Use with MECC 675 disk.

MATERIALS NEEDED

1. Copies of High Resolution sheet and grid, & Low Resolution sheet and grid for each student

2. Rulers and colored pencils for individual activity

VOCABULARY

graphics, low resolution, high resolution, coordinates, GR, COLOR=, PLOT, HLIN, VLIN, HGR, HCOLOR=, HPLOT

LESSON DESCRIPTION

1. Define graphics. Discuss graphics students have experienced in prepared programs. Discuss other applications of computer graphics.

2. Review Low Resolution Graphics sheet. Students may enter and RUN examples.

3. Do sample programs, together, on the board. Students should RUN the programs after drawing various parts to check for errors.

4. Review together High Resolution Graphics sheet.

5. Do a group High Resolution program on the board, students entering and running the program as it progresses.

6. Output a hard copy of the high resolution pictures on the printer.

INDEPENDENT ACTIVITIES

Students may draw their initials, a figure, message, or picture on graphics grid. Add coordinates. Code a program and run it. Output high res. on printer.

LOW RESOLUTION GRAPHICS

GR switches the system into low resolution graphics mode. GR will clear the screen for plotting. The screen grid is: X = 0 to 39, left to right on the screen. Y = 0 to 39 top to bottom on the screen.

COLOR= is a variable to which is assigned a value from 0 to 15. There are 16 colors in low resolution graphics. Black blends with the screen background so doesn't show. The color will remain in effect until COLOR= is changed.

Valid colors are:

0 - black	4 - dark green	8 - brown	12 - green
1 - red	5 - grey	9 - orange	13 - yellow
2 - dark blue	6 - medium blue	10 - grey	14 - aqua
3 - purple	7 - light blue	11 - pink	15 - white

PLOT statement is used to plot a single brick-shaped area on the display screen in the 40 by 40 grid. The color of the brick is determined by the value of the COLOR= statement. PLOT X,Y will plot a brick at coordinates (X,Y).

For example:

```
10 GR
20 COLOR=9
30 PLOT 20,20
```

will plot an orange brick in the middle of the screen.

HLIN statement is used to plot a low resolution <u>horizontal</u> line of bricks. 'HLIN X1,X2 AT Y' will draw a horizontal line from the beginning, X1, to the end, X2, at the height of Y.
For example:

```
10  GR
20  COLOR=11
30  HLIN 0,39 AT 20
```

will draw a pink line all the way from the left side of the screen to the right at just about halfway down the screen.

VLIN statement operates to produce a <u>vertical</u> line. 'VLIN Y1,Y2 AT X' will plot a vertical line from Y1 up or down the screen to Y2 at X spaces from the left side of the screen.
For example:

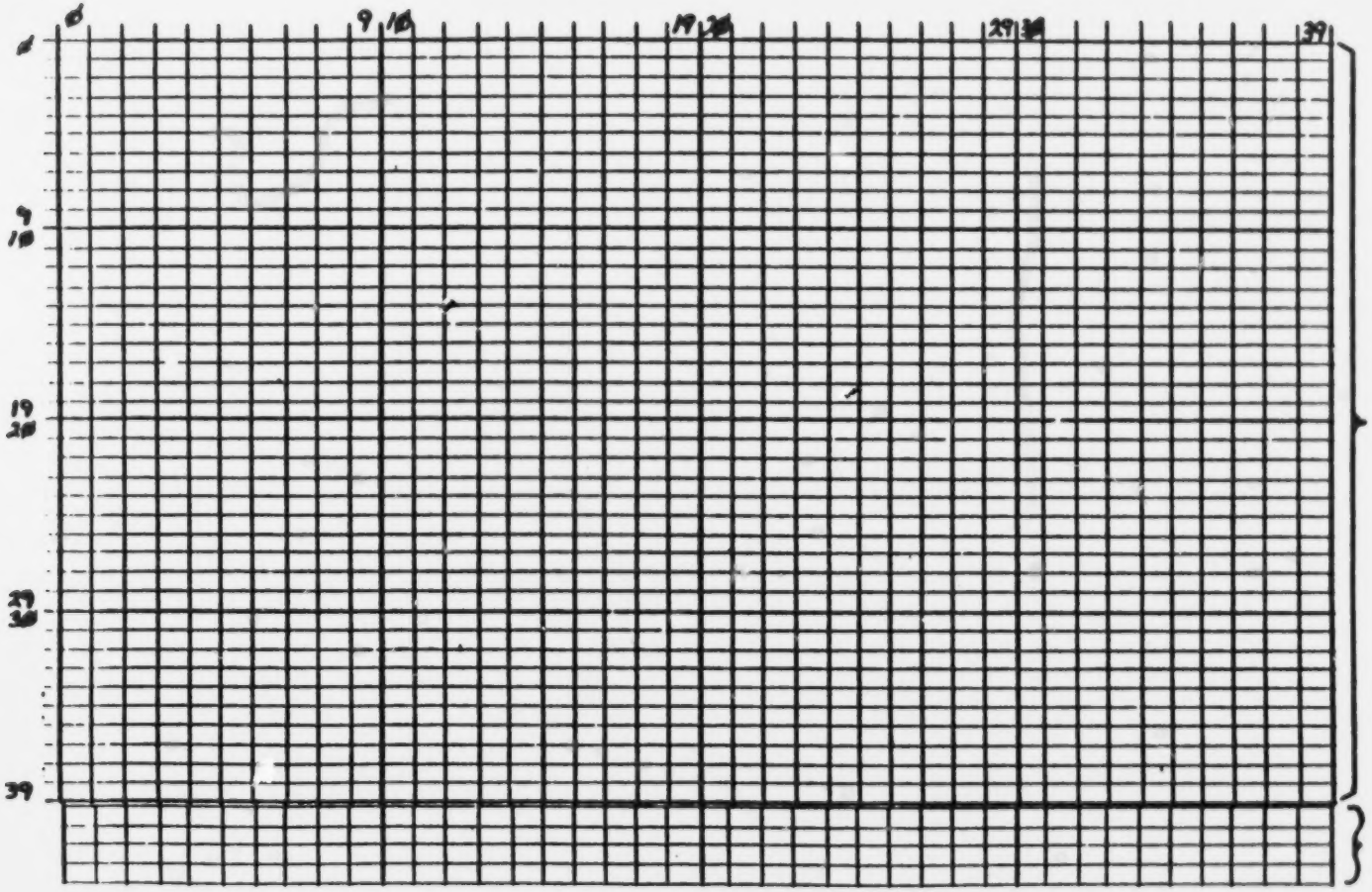
```
10  GR
20  COLOR=1
30  VLIN 20,39 AT 30
```

will draw a red line from halfway down the screen to the bottom just 10 spaces from the right of the screen.

TEXT takes you out of graphics mode, leaving a screen of random characters.
HOME will clear the screen and put the cursor at the top.
LIST will display your program.
RUN will execute your graphics program.

It is very difficult to print low resolution pictures.

Diagram of Output Screen For Applesoft Low-Resolution Graphics



MECC:80

BEST COPY AVAILABLE

HIGH RESOLUTION GRAPHICS

HGR switches screen display into high resolution graphics mode with a 4 line text window at the bottom of the screen. The possible screen coordinates are: X = 0 to 279, left to right, and Y = 0 to 159, top to bottom of the screen.

HGR2 utilizes the entire screen for graphing. The possible screen coordinates are: X = 0 to 279, left to right, and Y = 0 to 279, top to bottom of the screen.

HCOLOR= is a variable used for color selection. A range of 8 colors is available, numbered 0 to 7. The color will remain until HCOLOR= is changed.

0 - black	2 - purple	4 - black	6 - blue
1 - green	3 - white 1	5 - red	7 - white 2

NOTE : There is a difficulty with plots and vertical lines in high resolution. To avoid the problem, always plot even numbered colors on even numbered X-coordinates and odd colors on odd X-coordinates. Plotting an odd numbered color on an even X or visa-versa results in a black line which cannot be seen, i.e. your line disappears! There is no problem with Y.

HPLOT X,Y will plot a single point at the coordinates (X,Y). X has a range from 0 to 279. The Y coordinate varies from 0 to 159 or, if the entire screen is being used, from 0 to 191.

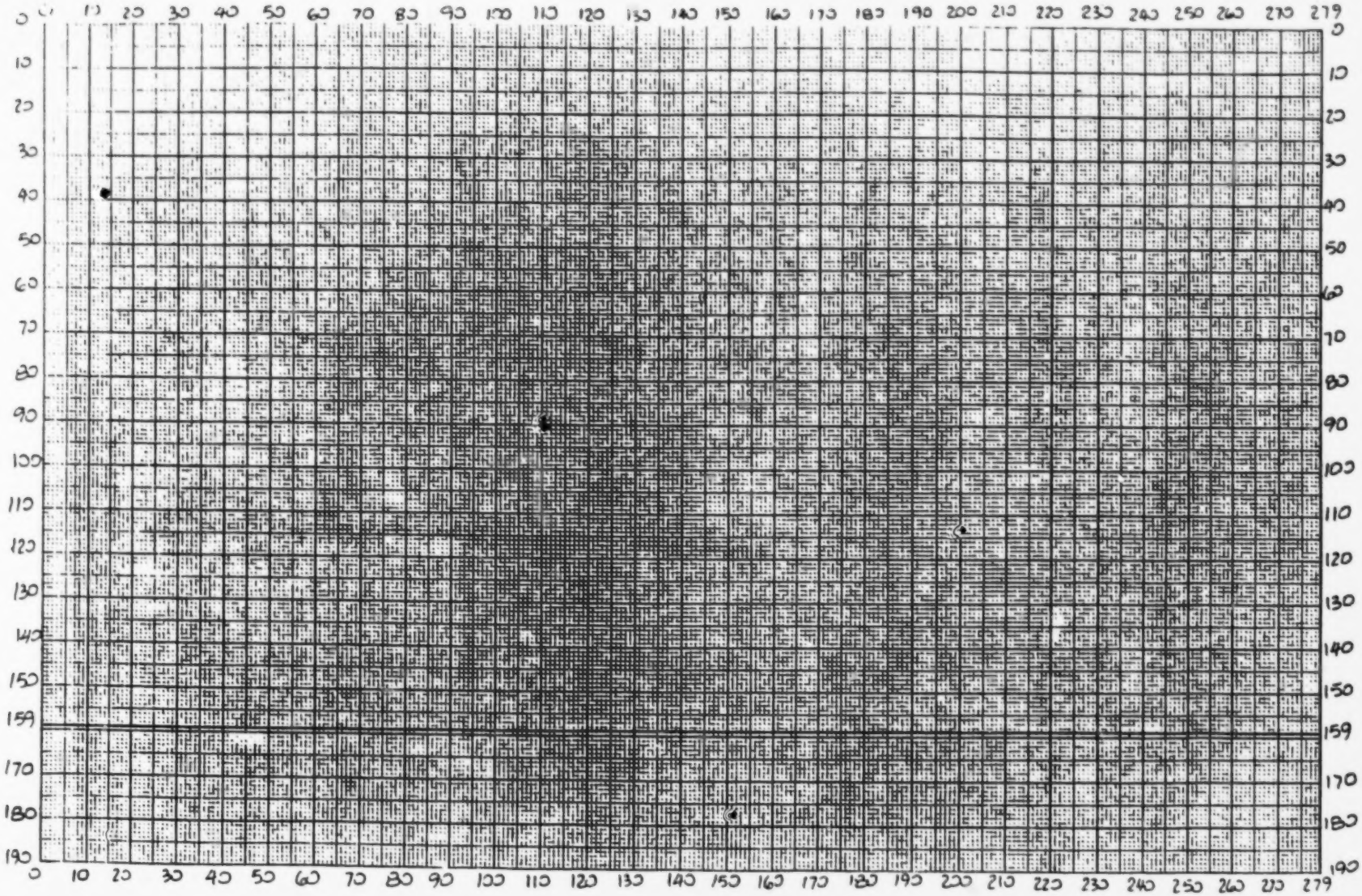
HPLOT X1,Y1 TO X2,Y2 TO X3,Y3 draws two connecting lines, from point (X1,Y1) to (X2,Y2) and from (X2,Y2) to (X3,Y3). Their color is determined by the HCOLOR= variable. Several connecting lines can be programmed on one BASIC line, using this statement.

HPLOT TO X,Y draws any line from the last point of the previous line to the point specified after TO. The color of the line will be that of the last point plotted, not that of the most recent value of HCOLOR=.

To print out high res graphics with a Grappler Plus interface card:

Control-I
G (graphics) will print single size picture.
Adding 2 (second graphics page) prints HGR2 (full screen) pictures.
Adding D (double) will make the picture double size.
Adding E (emphasize) will make it darker.
Adding R (rotate) will rotate picture 90 degrees.
Adding I (inverse) will print white on black.
Note: None of the above should print on the screen. If it prints - try again.

Example: PR#1 RETURN CONTROL-I GDE RETURN will print large and dark.



GRAPHICS EXAMPLES

Let's draw a face in LOW RESOLUTION GRAPHICS.
RUN it after you plot nose, eyes, mouth, etc.
To get back to the listing: push CONTROL-RESET and type LIST.

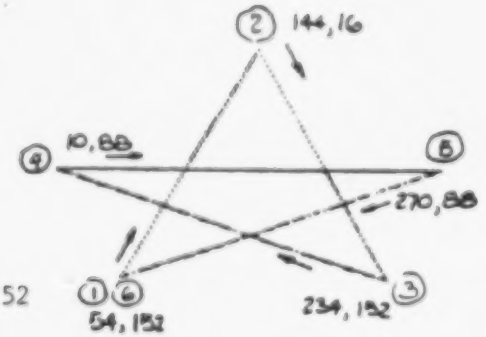
```
NEW

10 GR                          10 GR puts the computer in low res. graph. mode.
20 COLOR= 9                    20 Select a color for the nose.  9 is orange.
30 PLOT 20,20                  30 Nose - 20 spaces across, 20 spaces down.
40 COLOR= 2                    40 Choose blue for the eyes?
50 PLOT 10,10                  50 Left eye - 10 from left, 10 from top
60 PLOT 30,10                  60 Right eye - 30 from left, 10 down
70 COLOR= 1                    70 Red for mouth
80 HLIN 10,30 AT 30            80 Horizontal line from 10 to 30, 30 spaces down
90 PLOT 9,29                   90 Let's make him smile - left side
100 PLOT 31,29                 100 Right side smile
110 PLOT 8,28                  110 Left side - more smile
120 PLOT 32,28                 120 Right side - more smile
```

Do you want to put a frame around the picture? If so, add these lines.
Starting at the top left corner -

```
130 COLOR= 15                  130 White, but your choice
140 HLIN 0,39 AT 0             140 Left to right at top
150 VLIN 0,39 AT 39            150 Top to bottom at far right
160 HLIN 39,0 AT 39            160 Right to left at bottom
170 VLIN 39,0 AT 0             170 Bottom to top at left
```

Save graphics the same as any BASIC program. Name it Face? Type SAVE FACE.

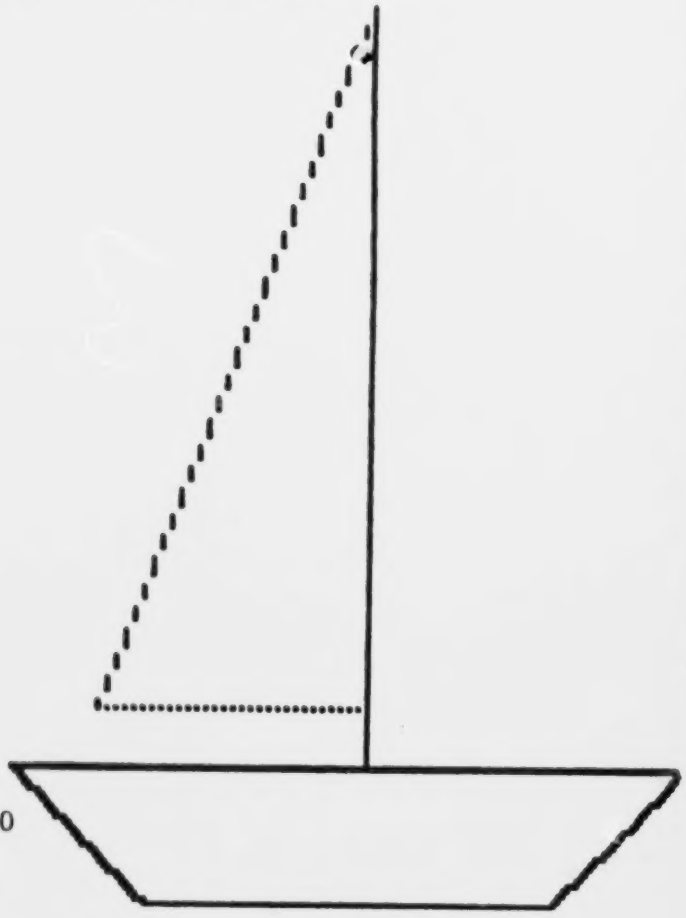
How about a star in HIGH RESOLUTION GRAPHICS:




```
NEW

10 HGR
20 HCOLOR= 2
30 HPLOT 54,152 TO 144,16
40 HPLOT 144,16 TO 234,152 TO 10,88 TO 270,88 TO 54,152
```

Explanation:
10 - HGR puts the computer into high resolution graphics mode.
20 - HCOLOR= sets a color which remains until changed.
30 - HPLOT...TO draws a line from the first coordinates to the second.
40 - Starts at the end of the first line and draws all the remaining, connected lines, back to the beginning.

HIGH RESOLUTION GRAPHICS CLASS PROJECT: Draw a sailboat.



PROGRAM

```
10 HGR2
20 HCOLOR= 6
30 HPLOT 140,20 TO 140,160
40 HCOLOR= 3
50 HPLOT 141,20 TO 81,150 TO 141,150
60 HCOLOR= 1
70 HPLOT 61,160 TO 211,160
80 HPLOT 61,160 TO 91,185 TO 181,185 TO 211,160
90 END
```

COMMENT

10 HGR2 will use the entire screen for printing. There will be no text window.
20 We will start with the mast. Color chosen is blue.
30 Top of mast is halfway across the screen (140) and 20 spaces down.
Plot mast from top to bottom.
40 Change color to white for sail. NOTE: Because we now have an odd numbered color, we will make the X part of the X,Y coordinates, odd values. We do this by adding 1 to the X value.
50 Plot sail from top of mast, left to the tip, and continue back to the mast.
60 Change color to green (7) for boat. Note odd color so odd X's to follow.
70 Draw the top of the boat.
80 Draw the rest of the boat from left to right.
90 Optional END

PRINTING HGR2 PICTURES

Turn on the printer with PR#1 RETURN.
To output an HGR2 picture on the printer, add a 2 to the code.

PR#1 RETURN CONTROL-I GDE2 RETURN output the above picture.

SAMPLE (APPLE) BASIC PROGRAMMING TEST

1. In order to output a character to the screen, you must use the keyword
   A. HOME B. NEW C. PRINT D. GOTO

2. To prevent a file from being accidentally erased from the disk, the following command is used:
   A. LOCK B. DELETE C. SAVE D. LOAD

3. Branching is
   A. a flowchart symbol B. transfering control to a different line number
   C. a collection of unorganized data D. a series of instructions

4. Every program line starts with a
   A. keyword B. statement C. number D. semi-colon

5. The keyword that assigns a color in low resolution graphics is
   A. COLOR B. COLOR= C. HCOLOR= D. GR

6. To erase the computer's RAM memory, use
   A. DELETE B. HOME C. CLEAR D. NEW

7. The keyword that draws a dot in high resolution graphics is
   A. HPLOT B. VLIN C. HLIN D. HGR2

9. The keyword that draws a vertical line in low resolution graphics is
   A. HPLOT B. VLIN C. HLIN D. COLOR=

10. A sequence of program instructions that are repeated a given number of times is called
    A. a shunt B. a variable C. a constant D. a loop

11. The keyword that sets high resolution graphics mode with a text window is
    A. PLOT B. HPLOT C. HGR D. HGR2

12. To erase a file from a disk use
    A. DELETE B. ERASE C. UNLOCK D. SAVE

13. The program that runs the operation of the disk is the
    A. Master B. ROM C. DOS D. CPU

14. To prepare a blank disk for use it must be
    A. initialized B. saved C. unlocked D. pre-filed

15. The disk used to copy and initialize disks is the
    A. Verifying program B. Slave C. System Master D. Initializer

16. Which of the following will draw a brick of color in low resolution graphics?
    A. PLOT B. HPLOT C. HLIN D. VLIN

Rewrite the following lines so they are correct:

17. 20 PRINT TODAY IS THE BEGINNING OF THE REST OF MY LIFE.

18. 10 WRITE 45/5+2

19. 60 PRINT 3(5)

20. HPLOT 44,22 AT 68,40

Write the output from the following program lines.

21. PRINT 5 * 5

22. PRINT "FOX"

23 PRINT "6/3"

Match the following.

____	24.	Clears the screen and puts the cursor at the top	A. DELETE
____	25.	Erases memory	B. RENAME
____	26	Changes the name of a program on the disk	C. LOAD
____	27.	Causes the program to branch to a specified line	D. SAVE
____	28.	Displays all or part of the program in memory	E. HOME
____	29.	Transfers a program from disk to RAM	F. LOCK
____	30.	Transfers a program from RAM to disk	G. NEW
____	31.	Sets high resolution graphics mode with text window	H. LIST
____	32.	Erases a file from the disk	I. GOTO
____	33.	Prevents a file from accidental erasure	J. HGR

Sample BASIC Programming Test

Answer Key

1. C.
2. A
3. B
4. C
5. B
6. D
7. D
8. C
9. B
10. D
11. C
12. A
13. C
14. A
15. C
16. A
17. Add quotes before TODAY and after LIFE.
18. change WRITE to PRINT
19. 3 * 5
20. change AT to TO
21. 25
22. FOX
23. 6/3
24. E
25. G
26. B
27. I
28. H
29. C
30. D
31. J
32. A
33. F

LESSON NUMBER	23
TITLE	Programming in BASIC - 4: Memory and variables, LET, READ...DATA, INPUT
GOAL	Programming Logic : Students will understand and apply simple logic, principles and techniques related to computer programming.
LEARNING OUTCOMES	1. Students will know the difference between a variable and a constant. 2. Students will understand that numeric variables store numbers. 3. Students will know that variables may be single letters, double letters or single letters followed by single numbers. 4. Students will understand that the LET and READ...DATA statements assign values to variables from within the program. 5. Students will understand that the INPUT statement allows the user to assign values to variables. 6. Students will understand how to use string variables. 7. Students will experience the use of variables in prepared programs.
BACKGROUND INFORMATION	This lesson should reinforce student knowledge of computer storage operation. It will also demonstrate how interactive programs, which the learner has worked with in the software section, are created.
REFERENCE MATERIALS	1. MECC Beginning Applesoft Basic , pages III-6 to III-13, pages V-1 to V-6, pages IV-1 to IV-7 2. A Guide to Programming in Applesoft

MATERIALS NEEDED	1. Attached sheets on LET, READ...DATA, and INPUT 2. Overhead or handout of MECC III-11
VOCABULARY	memory, variable, constant, numeric variable, string variable, LET, READ...DATA, INPUT, REENTER, comma, semi-colon, prompting message, user defined variable
LESSON DESCRIPTION	1. Discuss and define the difference between a variable and a constant. 2. Discuss the rules for variables using MECC III-11 overhead or handout. 3. Give examples of variables that are double letter or letter and number. 4. Using examples, show how the LET statement is used to store information in the computer's memory. Have the computer PRINT what was stored. 5. Discuss examples of when READ...DATA would be used to enter data into memory. Note the ease of adding more DATA statements when needed. 6. Describe the INPUT statement and how and why it is used. Enter and RUN the examples in the INPUT sheet.
INDEPENDENT ACTIVITIES	Assign MECC activity page III-17, activities 4 & 5.

STORING DATA IN MEMORY

Memory can be thought of as a series of information storage locations. Each storage location, used, is named by a variable. A variable is a symbol that may assume or store many different values. The computer uses variables to represent numbers or strings.

Numeric variables

If the variable is a numeric variable, it can only hold a number. This number can be used for math, etc. Numeric variables are represented by a letter, 2 letters, or a letter and a number.

Examples are: A, YY, X1, B2, etc.

String Variables

If the variable is a string variable, then alphanumeric characters - letters, numbers, and special symbols, can be stored there. String variables are represented the same as numerics except the variable is immediately followed by a dollar sign ($) often pronounced 'string'.

Examples are: A$, YY$, X1$, B2$, etc.

NOTE: You cannot do mathematical operations on string variables. If X$="246", you cannot do X$/2. Whereas you can with numerics. If X=246, X/2 is 123.

In the following model of information stored in memory, can you see why the memory locations were named as they were?

contents	BLOGGS	JOE	16	JUNIOR	1986	NORWALK	CT	06854
variable	LN$	FN$	A	C$	GY	C$	S$	Z

Variables can take on different values in the same program.
Try these sample programs.

In the first one, X and Y are numeric variables. Each names a separate memory location where the number is stored. Line 10 assigns the value, 12, to memory location X. Line 40 assigns a new value to X. The contents of X is then 15. The value of Y also changes.

The second sample shows use of string variables. This time B$ takes on two different values. Which one will be printed by line 60?

```
NEW
10 X = 12
20 Y = 2 * X
30 PRINT X,Y
40 X = 15
50 Y = 3 * X
60 PRINT X,Y
```

```
NEW
10 B$ = "GEORGE"
20 PRINT B$
30 B$ = "JUDY"
40 PRINT B$
50 PRINT
60 PRINT "IT IS NICE TO SEE YOU, ";B$
```

LET

LET assigns a value to a variable. The value can be a constant (a number), a character string, or it can be an expression which will be evaluated before the assignment is made. A variable is a name that represents a value, stored in memory. The LET statement consists of a variable, an equals sign (=), and a value or formula. The word LET is optional. LET X = 1 is the same as X = 1. All values assigned to string variables must be enclosed in quotes.

Enter and RUN this example:

```
5' REM   PRINT A CASHIER'S BILL
8  REM   Q = QUANTITY    I$ = ITEM    P = PRICE    A = AMOUNT TO BE PAID
10 LET Q = 2
20 LET I$ = "RECORD ALBUMS"
30 LET P = 8.50
40 PRINT "QUANTITY", "ITEMS","PRICE"
50 PRINT Q, I$, P
55 LET A = Q * P
60 PRINT
70 PRINT "PLEASE PAY  $"; A
```

In lines 10, 20, and 30, LET assign values to variables.
Line 40 prints headings.
Notice: commas in a PRINT statement divide the output into 3 fixed columns.
Line 50 prints the values of the variables, again in columns.
Line 60 prints a blank line.
Line 70 prints the message, the dollar sign, and the value of A.
Note: A semi-colon in a PRINT statement tells the computer to print the next characters on the same line, up close, with no spaces in between.

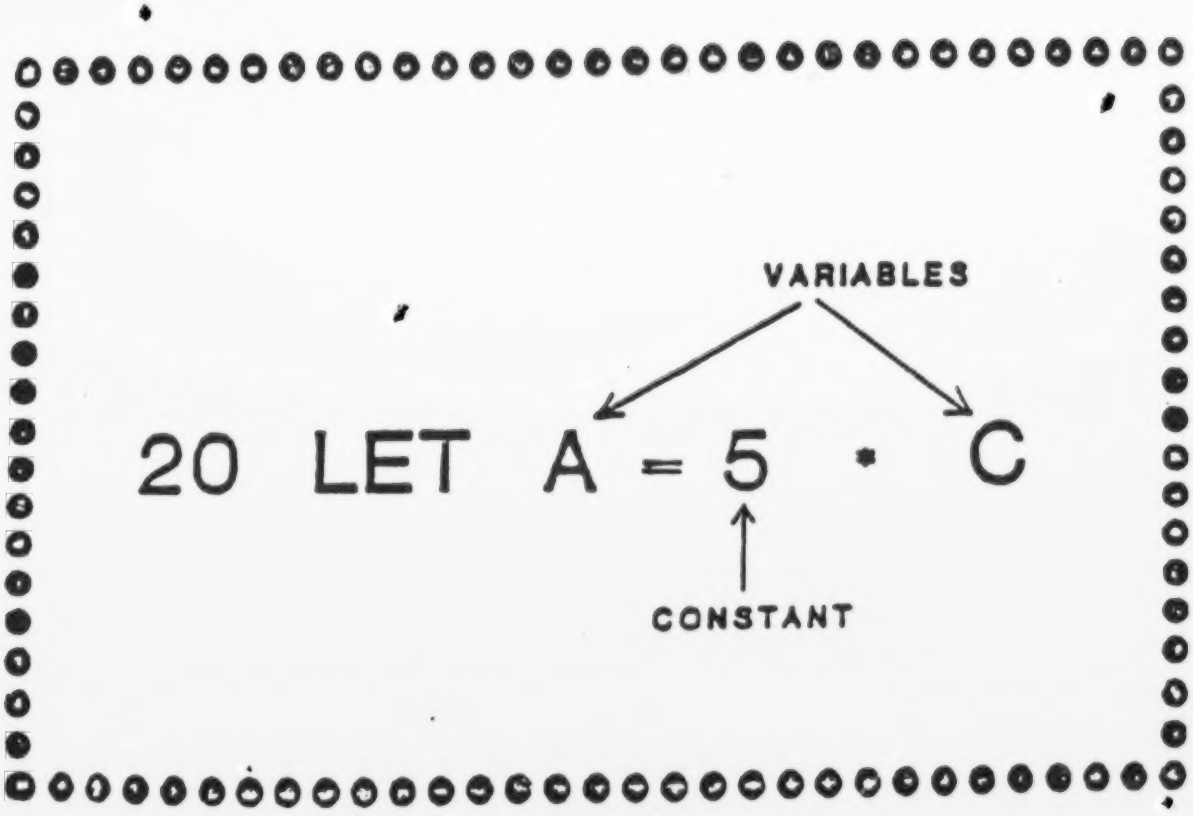
READ...DATA

READ and DATA are two statements, used together, that assign values to variables. The DATA statement lists the data, which are assigned, from left to right, to the variables listed in the READ statement. Fractions or arithmetic calculations like 4*3 are not allow in the DATA statement. Data statements are usually found at the end of the program. The GOTO statement allows the variables in the READ statement to be used more than once by creating a loop. Commas are always used to separate variables in READ and values in DATA.

Example:

```
10 READ N$, A, SS$
20 PRINT N$, A, SS$
30 GOTO 10
40 DATA JONES, 16, 042-13-4426, HARTMAN, 18, 041-57-0184
50 DATA NUNEZ, 15, 043-11-5439

RUN
JONES                16              042-13-4426
HARTMAN              18              041-57-0184
NUNEZ                15              043-11-5439
OUT OF DATA ERROR IN 10
```

The error message is to be expected and just says that the computer found no more data to process.



When using variables and constants in a LET statement, several rules must be observed. If variables and constants ane not used correctly, the computer cannot understand the instructions or execute the program.

1. One <u>variable</u> must be placed on the left side of the equal sign.

2. <u>Constants</u> may <u>not</u> be used on the left side of the equal sign.

3. As many constants and variables as necessary may be used on the right side of the equal sign.

INPUT

The INPUT command accepts data or information from the keyboard and assigns it to one or more variables.
Used alone, INPUT presents the user of the program with a question mark and waits for his/her response. It is good practice to <u>always</u> output a prompting message informing the user what information to type in.

Example:

```
10 REM     PRACTICE INPUT
20 PRINT "WHAT IS YOUR NAME"
30 INPUT N$
40 PRINT "PLEASE ENTER THE NAME OF YOUR SCHOOL"
50 INPUT S$
60 PRINT "HELLO, ";N$,".  ";S$;" IS A GREAT SCHOOL!"
70 END

RUN
WHAT IS YOUR NAME                                        (line 20)
?        (user types - SARAH)                            (line 30)
PLEASE ENTER THE NAME OF YOUR SCHOOL                     (line 40)
?        (user types - PODUNCK HIGH)                     (line 50)
HELLO, SARAH.   PODUNCK HIGH IS A GREAT SCHOOL!          (line 60)
```

INPUT can be used with numerics as well as strings. Try this program.

```
10 REM  FIND YOUR GRADE AVERAGE
20 PRINT "ENTER FIVE GRADES, SEPARATED BY A COMMA OR RETURN"
30 INPUT A,B,C,D,E
40 X = (A+B+C+D+E) / 5
50 PRINT "YOUR GRADE AVERAGE IS "; X
60 END
```

If the user presses RETURN after each grade a double question mark (??) will prompt each time for more data.

If values are being assigned to numeric variables and the user inputs a string, a REENTER (re-enter) error message will appear. Only numbers can be assigned to numeric variables. String variables will accept letters or numbers.

There is a shortcut which combines INPUT and the PRINT message into one line. Let us try it using the first practice example above.

```
10 REM    COMBINE INPUT AND PRINT
20 INPUT "WHAT IS YOUR NAME "; N$
30 INPUT "PLEASE ENTER THE NAME OF YOUR SCHOOL "; S$
40 PRINT "HELLO, ";N$;".  ";S$;" IS A GREAT SCHOOL!"
50 END

RUN
WHAT IS YOUR NAME (user types SARAH here)
PLEASE ENTER THE NAME OF YOUR SCHOOL  (user types - PODUNCK HIGH)
HELLO, SARAH.  PODUNCK HIGH IS A GREAT SCHOOL!
```

LESSON NUMBER 24

TITLE Programming in BASIC - 5: IF...THEN, FOR...NEXT

GOAL Programming Logic : Students will understand and apply simple logic, principles, and techniques related to computer programming.

LEARNING OUTCOMES

1. Students will know the functions of the comparison symbols <, <=, >, >=, =, <>.

2. Students will know how to use the IF...THEN statement to PRINT, GOTO, LET, or END, based on a condition being true or false.

3. Students will use the FOR...NEXT statements to execute a set of instructions a given number of times.

4. Students will understand the distinction between an endless and a controlled loop.

BACKGROUND INFORMATION

It is not expected that the average Computer Education class will progress to this lesson in programming. It has been included because these keywords and control concepts complete an elementary overview of BASIC programming. The exceptional student should be exposed to this lesson for this reason.
Flowcharts would be a useful tool in visualizing these loops.

REFERENCE MATERIALS

1. MECC Beginning Applesoft Basic , pages IV-12 to IV-18 and pages V-7 to V-12.

2. A Guide to Programming in Applesoft , pages 2.1 to 2.4 and pages 2.7 to 2.10.

MATERIALS NEEDED

Copies of IF...THEN and FOR...NEXT sheets to all students.

VOCABULARY

comparison symbols, bottom of loop, conditional statement, IF...THEN, controlled loop, FOR...NEXT

LESSON DESCRIPTION

1. Describe and define the symbols for less-than, less-than or equal to, greater-than, greater-than or equal to, equal to, and unequal to.

2. Disscuss situations where comparisons must be made.

3. Review IF...THEN sheet with students. Enter, RUN, and discuss the example program.

4. Review GOTO and the endless loop. Discuss the applications of a controlled loop.

5. Review FOR...NEXT sheet with students. Enter, RUN, and discuss examples.

INDEPENDENT ACTIVITIES

Assign MECC activity page IV-18 activities 3, 5, 6, and 7.

IF...THEN

The IF...THEN statement is a conditional statement. It allows the computer to take action or not, based on a condition. If the condition is met, or true, then the statement is executed. If the condition is not met, or false, then the entire IF...THEN statement is ignored by the computer.

The construction of the statement is: line# IF [condition] THEN [statememt]. BASIC statements GOTO, PRINT, LET, INPUT, or END can follow THEN.

The IF part of IF...THEN is a logical comparison. Two quantities are compared using the following symbols:

=	equal to
<>	unequal to
>	greater than
<	less than
>=	greater than or equal to
<=	less than or equal to

EXAMPLE:

```
10  REM  PRACTICE IF...THEN
20  LET X = 0
30  LET X = X + 1
40  PRINT "X = ";X
50  IF X = 5 THEN PRINT "GOOD"
60  IF X = 6 THEN LET X = 23
70  IF X / 5 = 5 THEN PRINT "NOW X MUST = 25!"
80  IF X = 15 THEN PRINT "BOO HOO"
90  IF X = 28 THEN GOTO 150
100 IF X = 30 THEN PRINT "THE END"
110 IF X > 29 THEN END
120 GOTO 30
130 :
140 REM THIS SECTION NOT IN THE LOOP
150 PRINT "ALMOST THE END"
160 GOTO 30
```

ANALYSIS BY LINE #

```
10 Explanatory REMark.
20 Initialize counter to 0.
30 Add 1 to counter each loop.
40 Print the value of X.
50 Conditional PRINT statement.
60 Cond.  Assign new value to X.
70 Expression evaluated in Cond.
80 Not executed. X never = 15.
90 Transfer control to line 150.
100 Conditional PRINT.
110 Stops program when X > 29.
120 Bottom of loop.
130 : Empty colon - see below.
140 Explanatory remark.
150 Only printed if X = 28.
160 Gets back into loop.
```

Like REM, an <u>empty colon</u> (:) is ignored by the computer. It is used to separate sections of the program listing - to make it more readable.
A <u>colon</u> can also be used to put more than one statement on a program line.

For example:

```
40 PRINT "GOOD MORNING" : GOTO 110
```

Whenever possible, however, it is better to avoid more than one statement per line as it makes reading and debugging (finding mistakes) more difficult.

FOR...NEXT

FOR and NEXT statements are used to establish a controlled loop which will repeat a set of statements a given number of times. FOR and NEXT always appear as a pair. The FOR statement sets the variable to count from the initial value to an ending value. The NEXT statement, the bottom of the loop, increments the value of the variable by 1. NEXT then compares that variable value to the ending value to see if the loop is finished. When the ending value is reached, the loop will stop and control will pass to the line after NEXT.

The form of the statements is:

```
line # FOR [variable] = [initial value] TO [ending value]
       ) statements to be executed
line # NEXT [same variable]
```

Examples:

```
10 FOR Y = 1 TO 5
20 PRINT "PRINT 5 TIMES"
30 NEXT Y
40 END
```

'PRINT 5 TIMES' will be printed 5 times.

```
10 FOR T = 0 TO 10
20 PRINT T
30 NEXT T
40 END
```

T is incremented at each NEXT until T = 10. 0 thru 10 is printed.

Enter and RUN this program.

```
 10 HOME
 20 FOR X = 1 TO 4
 30 PRINT
 40 PRINT "THE VALUE OF X IS NOW ";X
 50 PRINT "DOES X = 4 YET?  ";
 60 IF X = 4 THEN PRINT "YES!!"
 70 IF X <> 4 THEN PRINT "NO"
 80 NEXT X
 90 PRINT
100 PRINT "END OF PROGRAM"
110 END
```

Explanation by line number:

```
10 Clear the screen. Put cursor at top.
20 Assign 1 to X. Loop exactly 4 times.
30 Print a blank line.
40 Print out the present value of X.
50 ; at end = next chars. on same line.
60 Conditional PRINT statement.
70 Conditional PRINT statement.
80 Bottom of loop. Compares X to 4.
90 Executed after loop is done.
100 Message to user.
110 Ends the program.
```

STEP # can be added to the FOR statement to increment the loop by a value other than 1.

This program counts by 5.

```
10 FOR Z = 0 TO 100 STEP 5
20 PRINT Z
30 NEXT Z
```

This program counts backwards by 5.

```
10 FOR Q = 100 TO 0 STEP -5
20 PRINT Q
30 NEXT Q
```

GLOSSARY

abacus
: an ancient calculating device consisting of a wooden frame with rods on which beads are moved

analog computer
: a computer that measures continuously changing conditions, such as temperature and pressure, and converts them into quantities

algorithm
: a step-by-step plan or procedure for solving a problem

Analytical Engine
: a mechanical computing device designed, but never completed, by British scientist Charles Babbage in 1938. It was the forerunner of modern digital computers

arithmetic logic unit (ALU)
: one of the two main parts of the central processing unit; does calculations and compares numbers

Arithmetic Machine
: a calculation machine that could add and subtract; invented by Blaise Pascal in 1642

Assembly Language
: a symbolic language that has a one-to-one correspondence with machine language instructions

Babbage, Charles
: a British scientist who developed and partially built the Analytical Engine, which was the forerunner of modern digital computers

BASIC
: An acronym for Beginner's All-purpose Symbolic Instruction Code. A computer language used in most microcomputers and often used in mainframes and minicomputers

binary number system
: a number system that uses only two digits, 0 and 1

bit
: a binary digit, 0 or 1 used to represent the "off" or "on" state of an electric circuit. It is the smallest unit of digital information a computer processes

branching
: a technique used to transfer control from one line of a program to another

bug
: any mistake in a computer program

byte
: a unit made up of 8 bits. One byte represents one character on the computer keyboard

card reader
: an input device that transfers data from punched cards to the computer's memory

cassette tape
: a strip of magnetic material on which computer programs are stored in the form of magnetic impulses. Used with a tape recorder as an I/O device for microcomputers

cathode ray tube (CRT)
: see video screen

central processing unit (CPU)
: the part of a computer system that processes information. Its two main parts are the arithmetic logic unit and the control unit

character
: a letter, number, or symbol found on a computer keyboard

chip
: see integrated circuit chip

COBOL (Common Business Oriented Language)
: a computer language used for business applications

command
: a word (in BASIC) that usually tells the computer to do something with a program. For example, RUN is a command

compiler
: a computer program used to translate high-level language programs into machine-language programs, for example - FORTRAN

computer
: a machine designed to accept information, store information, process information, and give out processed information

computer engineer
: a person who designs circuitry used in computer systems

computer operator
: a person who runs a computer, particularly a mainframe

computer technician
: a person who repairs computers. A technician also works with an engineer in designing a computer

constant
: a quantity with a fixed value, a number

control unit
: one of the two main parts of the control processing unit. It directs the flow of information through a computer system

cursor
: the symbol appearing on the video screen that shows where the next character to be typed will appear

data
: the information that is to be input or that is the output of a program

data bank
: a collection of computerized information

data processing
: taking data, or information, that has been input and using it to produce information in other forms such as reports

decision symbol
: a flowchart symbol used at a decision point in the program

delete
: a BASIC command that erases a file from a disk

digital computer
: a general-purpose computer that uses letters, numbers, and symbols as input and converts this information into digits to be stored and processed

disk
: a flat, circular device with a magnetic surface that is capable of storing computer programs. Some disks are flexible (see floppy desk) and some are hard (see hard disk)

disk drive
: an I/O device that loads a program or data stored on a disk into a computer or from computer memory onto the disk

DOS (Disk Operating System)
: the programs that keep track of files, save and retrieve information and do a multitude of other housekeeping tasks

Eckert, J. Presper
an American engineer who, with John Mauchly, designed the ENIAC and UNIVAC computers

EFT (Electronic Fund Transfer)
the process of transferring funds from one bank account to another electronically

electromechanical
made with both electric and mechanical (moving) parts

electronic cottage
a system allowing employees to perform work or services at home via computers with telecommunications links

electronic mail
the process of sending messages from one computer or computer terminal to another

electronic publishing
information ranging from news, stockmarket reports, sports scores, catalog shopping and reference books available on a time-shared computer system

ENIAC (Electronic Numerical Integrator and Calculator)
the first electronic computer designed by Eckert and Mauchly in 1946

floppy disk
a magnetic storage device commonly used with microcomputers

flowchart
a pictorial representation of a step-by-step solution to a problem

FORTRAN (FORmula TRANslator)
a programming language commonly used for scientific and engineering applications

GOTO
a BASIC statement that causes program execution to branch to a specified line

graphics
non-text designs and patterns displayed as output

hard copy
paper output from a printer often called a printout

hard disk
a long lasting magnetic storage device typified by non-removable data, large storage and very fast transfer of a great amount of information

hardware
: computer machinery

high level language
: an English-like programming language. BASIC is an example

Hollerith, Herman
: an American engineer who designed the Tabulating Machine, which used punched cards to sort the data from the 1890 census

IF - THEN
: a BASIC statement that conditionally causes the program to execute an instruction

information retrieval
: one function of a computer that deals with sorting out requested information from a large set of stored data

initializing
: erases all files from a diskette and places a copy of DOS in tracks 0, 1, and 2

input
: information to be put into the computer

input device
: part of a computer system (usually a keyboard or storage device) that puts information into the computer's memory. It is sometimes referred to as an input unit

input/output symbol
: a flowchart symbol used when inputting data or printing an output

integrated circuit (IC)
: an electrical pathway, made up of many transistors, that transmits electricity much faster than a single transistor. Integrated circuits were used in third-generation computers

integrated circuit chip (ICC)
: a very thin wafer of silicon containing thousands of integrated circuits. Chips were used in fourth-generation computers

interpreter
: a program that executes input, program line by program line; does not generate a machine language program. Applesoft BASIC is an interpreter

I/O
: refers to input/output

Jacquard, Joseph
: a French weaver who, in 1801, designed a loom which used punched cards to store patterns that were woven into fabric

K (Kilobytes)
: term used to indicate memory size, such as "16K" (16,000 bytes). Technically 1K = 1024 bytes

keyboard
: an input device used for typing data to a computer

keypunch machine
: a machine used to punch holes in cards, which are used for computer input

keyword
: a control word such as PRINT, GOTO, or READ that gives an instruction to a computer

large scale integration (LSI)
: the process of putting thousands of integrated circuits on one silicon chip

LET
: keyword that assigns a value to a specified variable

light pen
: an input device, resembling a pen, that responds to light on the video screen

line number
: the number that tells the computer in what order to execute program instructions

LIST
: a BASIC command that displays all or part of the program currently in memory

LOAD
: a command that loads a program from cassette or disk into the RAM of the computer

LOCK
: protects a file from accidental erasure or change

loop
: a sequence of BASIC program instructions which are repeated until a predetermined terminal condition is achieved

machine language
: basic language of a computer consisting of binary digits

magnetic ink character recognition (MICR)
: a kind of input in which a MICR reader interprets characters printed with special magnetic ink

mainframe computer
: a large computer system that can handle many jobs at once

Mark I
: the first electromechanical computer. It was designed by Howard Aiken in 1944

Mauchly, John
: an American engineer who, with J. Presper Eckert, designed the ENIAC and UNIVAC computers

memory
: the part of a computer system that stores information

microcomputer
: a small computer that can perform one job at a time. It is also called a home or personal computer

Napier, John
: a Scottish mathematician who, in 1612, invented the calculation device known as Napier's Bones

Napier's Bones
: a set of numbered rods, developed by John Napier, which was used for calculating

NEW
: a BASIC command that deletes the current program and all variables from memory

numeric variable
: a variable beginning with a letter, to which a number is assigned

optical mark reader (OMR)
: an input device that reads pencil markings on paper

output
: the results of a processed program

output device
: the part of the computer system (usually a hard copy printer or video screen) that displays, prints, or records the results. It is sometimes referred to as an output unit

Pascal
: a computer language that was designed to be reliable and efficient with extremely quick error checking

peripheral equipment
: hardware, such as tape recorder or printer, that is connected to the main computer

plotter
: an output device that draws pictures and graphs on paper

PRINT
: a BASIC command that outputs characters to the screen or other output device

printer
: a device that prints output on paper

printout
: the output from a printer. It is also called hard copy

process symbol
: a flowchart symbol indicating actions to be taken

program
: a set of instructions, written in a computer language, that tells a computer what to do

prompt
: a symbol used by some computers to indicate when the computer is ready to accept input

punched card
: a paper card containing holes that stand for data programs

punched card loom
: weaving machine, perfected by Joseph Jacquard in 1801, which used punched cards to store patterns that were woven into fabric

query
: a language intended primarily to help non-programmers access a computer's data base

Random Access Memory (RAM)
: the computer's temporary memory that stores data and programs that are input from the keyboard or peripheral storage devices

Read Only Memory (ROM)
: the computer's permanent memory

READ - DATA
: the READ statement contains variables while the DATA statements store the values to be used by the variables

read/write head
: a device inside a disk drive that reads information on a disk. It can also "write" and erase information

reel-to-reel tape
: a kind of I/O storage

robot
: a computer controlled machine designed to do specific mechanical tasks

RUN
: a BASIC command that executes the program currently in memory

SAVE
: a BASIC command that saves the program currently in memory onto cassette or disk

scanner
: an input device usually used to read Universal Product Codes on packaged products

silicon
: a common element in the Earth's crust from which computer chips are made

simulation
: a computerized output which imitates a real-life situation

software
: programs for a computer. Usually refers to programs stored on devices such as disks or tapes

statement
: a word in BASIC that gives an instruction to the computer

string
: any combination of letters, numbers, or symbols (except quotation marks) within a BASIC program

string variable
: a variable assigned to represent letters, words and/or combinations of letters and numbers within a BASIC program

syntax error
: an error in the structural rules of a programming language. It is usually an error in spelling or punctuation

systems analyst
: a person who analyzes a situation and designs a computerized system to handle it

system master
: the diskette supplied with the computer that is used to initialize and copy diskettes and "boot" the system

Tabulating Machine
: a machine, designed by Herman Hollerith in 1887, which counted and recorded data from punched cards

tape drive
: an I/O device that reads and stores data on magnetic tape

tape recorder
: an I/O device commonly used with microcomputers

terminal
: an I/O device that has a keyboard for input and either a video screen or a printer for output. It is connected to a mainframe or minicomputer

terminal symbol
: a flowchart symbol that indicates a start or an end of a program

text
: refers to numbers, letters, or keyboard symbols (as opposed to graphics)

transistor
: a device used to transmit electricity in second-generation computers

translator
: a program that translates high level language into machine language

UNIVAC (UNIVersal Automatic Computer)
: It was designed by Eckert and Mauchly in 1951 and was the first commercial computer

Universal Product Code (UPC)
: a series of bars and numbers printed on packaged grocery goods. A UPC is read by a scanner in a computerized checkout system

UNLOCK
: removes locked status from a disk file, permitting it to be changed or deleted

vacuum tube
: a device used to transmit electricity in first-generation computers

variable
: a symbol that represents numerics or strings in memory

video screen
: an output device used to display information. It is also referred to as a cathode ray tube (CRT) or monitor

voice synthesizer
: an output device which translates computer output into human-like sounds

word processing
: the ability of a computer to allow the user to move or change words, sentences, and paragraphs without retyping them